

NASA Conference Publication 3255

1994 Space and Earth Science Data Compression Workshop

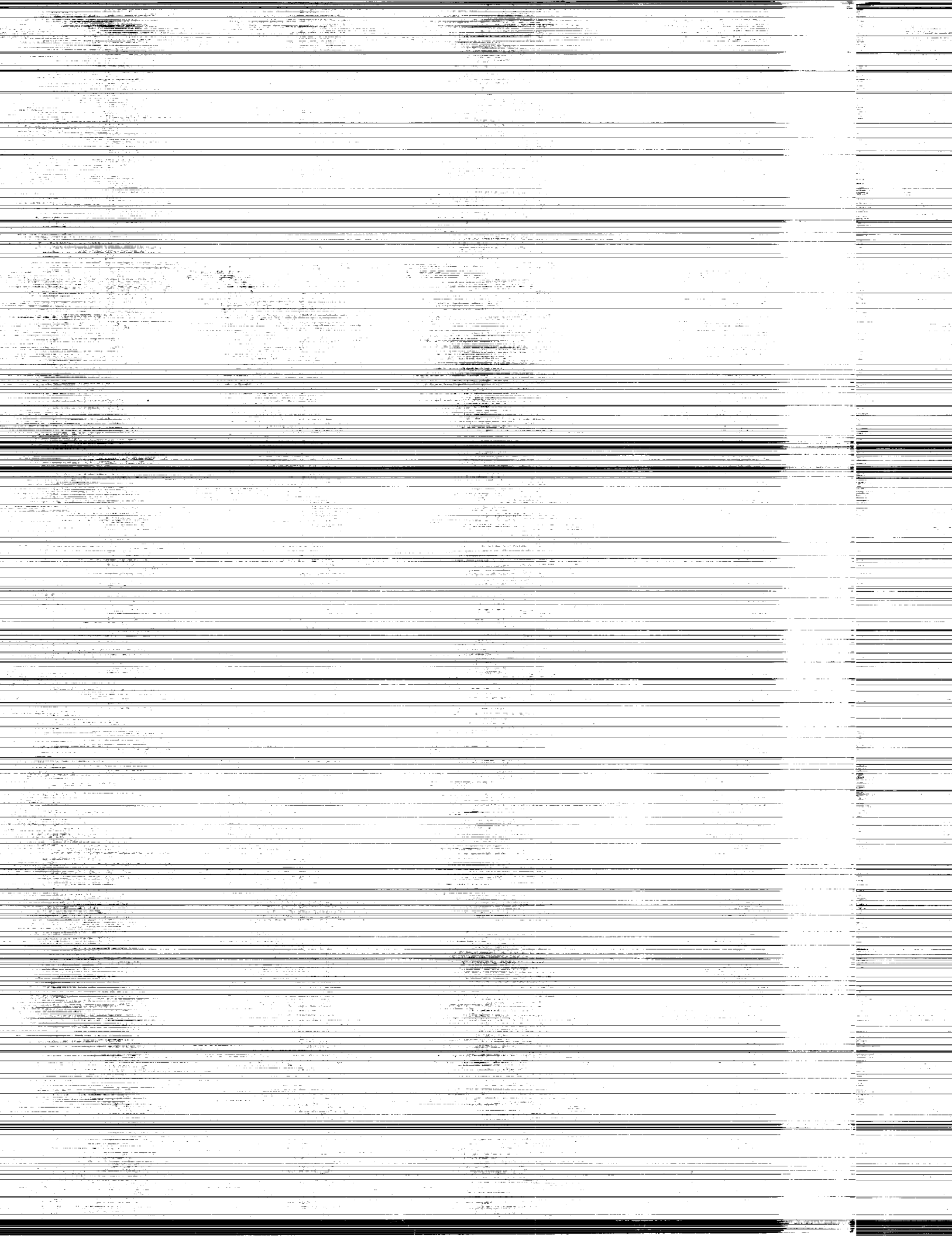
(NASA-CP-3255) THE 1994 SPACE AND
EARTH SCIENCE DATA COMPRESSION
WORKSHOP (NASA) 139 p

N94-28251
--THRU--
N94-28264
Unclass

H1/59 0000454

Proceedings of a workshop sponsored by the
National Aeronautics and Space Administration
at the Center of Excellence in Space Data and
Information Sciences and held at
the University of Utah
Salt Lake City, Utah
April 2, 1994





1994 Space and Earth Science Data Compression Workshop

Edited by
James C. Tilton
*NASA Goddard Space Flight Center
Greenbelt, Maryland*

Proceedings of a workshop sponsored by the
National Aeronautics and Space Administration
and the Center of Excellence in Space Data and
Information Sciences and held at
the University of Utah
Salt Lake City, Utah
April 2, 1994



National Aeronautics
and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

1994

FOREWORD

The fourth annual Space and Earth Science Data Compression Workshop was held on April 2, 1994, at the University of Utah in Salt Lake City, Utah. This NASA Conference Publication serves as the proceedings for the workshop. The workshop was held in cooperation with the 1994 Data Compression Conference (DCC'94), which was held at Snowbird, Utah March 29 - 31, 1994.

The goal of the Space and Earth Science Data Compression Workshop series is to explore the opportunities for data compression to enhance the collection and analysis of space and Earth science data. Of particular interest is research that is integrated into, or has the potential to be integrated into, a particular space and/or Earth science data information system. Participants are encouraged to take into account the scientist's data requirements, and the constraints imposed by the data collection, transmission, distribution and archival system.

Papers were selected from direct submissions to the Workshop and selected submissions to the 1994 Data Compression Conference (DCC '94). Thirteen papers were presented in 4 sessions. Discussion was encouraged by scheduling ample time for each paper.

The workshop was organized by James C. Tilton of the NASA Goddard Space Flight Center, Sam Dolinar of the Jet Propulsion Laboratory, Sherry Chuang of the NASA Ames Research Center, and Dan Glover of the NASA Lewis Research Center. Contact information is given below.

Acknowledgment

The organization of this workshop was supported by the Office of Advanced Concepts and Technology, NASA Headquarters, Washington, DC.

Workshop Organizers

James C. Tilton
Mail Code 930.1
NASA GSFC
Greenbelt, MD 20771
phone: (301) 286-9510
FAX: (301) 286-3221
Internet:
tilton@chrpisis.gsfc.nasa.gov
GSFCMAIL: JTILTON.

Sam Dolinar
Mail Stop 238-420
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
phone: (818) 354-7403
FAX: (818) 354-6825
Internet:
sam@shannon.jpl.nasa.gov

Sherry Chuang
NASA Ames Research Center
MS 269-4
Moffett Field, CA 94035-1000
phone: (415) 604-3376
Internet: chuang@ptolemy.arc.nasa.gov

Daniel Glover
Mail Stop 54-2
NASA Lewis Research Center
21000 Brookpark Road, Cleveland, OH 44135
phone: (216) 433-2847
Internet: caglove@lims01.lerc.nasa.gov

CONTENTS

Foreword	iii
Contents	v

PAPERS

An Image Assessment Study of Image Acceptability of the Galileo Low Gain Antenna Mission <i>S. L. Chuang and T. Grant, Ames Research Center;</i> <i>R. F. Haines and Yaron Gold, Recom Technologies; and</i> <i>Kar Ming Cheung, Jet Propulsion Laboratory</i>	3
Image Compression Software for the SOHO LASCO and EIT Experiments <i>Mitchell R. Grunes, Allied-Signal Technical Services;</i> <i>Russell A. Howard, Karl Hoppel and Stephen A. Mango, Naval Research Laboratory; and</i> <i>Dennis Wang, Interferometrics</i>	11
Comparison of the Lossy Image Data Compressions for the MESUR Pathfinder and for the Huygens Titan Probe. <i>P. Rüffer, F. Rabe and F. Gliem, Technische Universität Braunschweig; and</i> <i>H.-U. Keller, Max-Planck-Inst. f. Aeronomie</i>	23
Advanced End-to-End Simulation for On-Board Processing (AESOP) <i>Alan S. Mazer, Jet Propulsion Laboratory</i>	31
Performance Considerations for the Application of the Lossless Browse and Residual Model. <i>Walter D. Abbott III, Robert T. Kay and Ron J. Pieper, Naval Postgraduate School</i>	43
Image Quality Measures and Their Performance <i>Ahmet M. Eskicioglu, Paul S. Fisher and Siyuan Chen, University of North Texas</i>	55
A Comparison of Spectral Decorrelation Techniques and Performance Evaluation Metrics for a Wavelet-Based, Multispectral Data Compression Algorithm <i>Roy M. Matic, Hughes Research Laboratories; and</i> <i>Judy I. Mosley, Santa Barbara Research Center</i>	69
A Comparative Study of SAR Data Compression Schemes <i>C. Lambert-Nebout, CNES CT/AE/SEA/TB, O. Besson, ENSICA; and</i> <i>D. Massonnet and B. Rogron, CNES CT/IA/QTIS/PR</i>	81
Perceptual Compression of Magnitude-Detected Synthetic Aperture Radar Imagery <i>John D. Gorman and Susan A. Werness, Environmental Research Institute of Michigan</i>	93
Compression of Multispectral Landsat Imagery Using the Embedded Zerotree Wavelet (EZW) Algorithm. <i>Jerome M. Shapiro, Stephen A. Martucci and Martin Czigler,</i> <i>The David Sarnoff Research Center</i>	105
Vector Quantizer Designs for Joint Compression and Terrain Categorization of Multispectral Imagery <i>John D. Gorman, Environmental Research Institute of Michigan; and</i> <i>Daniel F. Lyons, The Mitre Corporation</i>	115

PRECEDING PAGE BLANK NOT FILMED

v

PAGE 14 INTENTIONALLY BLANK

Wavelet Compression Techniques for Hyperspectral Data	123
<i>Bruce Evans, Brian Ringer and Mathew Yeates, TRW Systems Integration Group</i>	
Some Practical Aspects of Lossless and Nearly-Lossless Compression of AVHRR Imagery.	135
<i>David Hogan, Atmospheric and Environmental Research, Inc.; and</i>	
<i>Chris Miller, Than Lee Christensen and Raj Moorti, Martin Marietta: Astro-Space Division</i>	

An Image Assessment Study of Image Acceptability of the Galileo Low Gain Antenna Mission

**S. L. Chuang⁽¹⁾ R. F. Haines⁽²⁾ T. Grant⁽¹⁾
Yaron Gold⁽²⁾ Kar Ming Cheung⁽³⁾**

SI-91

455

P. 8

**(1) Ames Research Center
Spacecraft Data Systems Res. Branch
Moffett Field, CA 94035**

**(2) Recom Technologies
Ames Research Center
Moffett Field, CA 94035**

**(3) Jet Propulsion Laboratory
Communications Systems Research
Pasadena, California 91109**

ABSTRACT

This paper describes a study conducted by NASA Ames Research Center (ARC) in collaboration with the Jet Propulsion Laboratory (JPL), Pasadena, California on the image acceptability of the Galileo Low Gain Antenna mission. The primary objective of the study is to determine the impact of the Integer Cosine Transform (ICT) compression algorithm (Cham, 1989) on Galilean images of atmospheric bodies, moons, asteroids and Jupiter's rings. The approach involved fifteen volunteer subjects representing twelve institutions involved with the Galileo Solid State Imaging (SSI) experiment (Belton et al., 1990). Four different experiment specific quantization tables (q-table) and various compression stepsizes (q-factor) to achieve different compression ratios were used. It then determined the acceptability of the compressed monochromatic astronomical images as evaluated by Galileo SSI mission scientists. Fourteen different images were evaluated. Each observer viewed two versions of the same image side by side on a high resolution monitor; each was compressed using a different quantization stepsize. They were requested to select which image had the highest overall quality to support them in carrying out their visual evaluations of image content. Then they rated both images using a scale from one to five on its judged degree of usefulness. Up to four pre-selected types of images were presented with and without noise to each subject based upon results of a previously administered survey of their image preferences. Fourteen different images in seven image groups were studied. The results showed that: (1) Acceptable compression ratios vary widely with the type of images; (2) Noisy images detract greatly from image acceptability and acceptable compression ratios; (3) Atmospheric images of Jupiter seem to have higher compression ratios of 4 to 5 times that of some clear surface satellite images.

INTRODUCTION

The Galileo spacecraft was launched in October 1989, and it will reach Jupiter and its moons in late 1995. Its mission includes Io flyby, releasing a probe into the Jovian atmosphere, probe data capture and relay, Jupiter orbital insertion, and 10 satellite encounters with Ganymede, Callisto, and Europa. In April 1991, when the spacecraft first flew by Earth, the Galileo team commanded the spacecraft to open the 1.8m X-band high-gain antenna (HGA), but it failed to deploy. The only way to communicate between Earth and the spacecraft is now through the use of one of the two S-band low-gain

antennas (LGA), which at Jupiter's range, can only support a telemetry data rate of 10 bit/second compared to the expected data rate of 134kbits/second in the HGA mode. Since the detection of the HGA anomaly, several unsuccessful attempts (including a major effort to perform hammering or pulsing of the deployment motor in December 1992) were made to free the HGA. A parallel effort was conducted from December 1991 through March 1992 to evaluate various options for improving Galileo's telemetry downlink performance in the event that the HGA would not open.

This contingency plan was known as the Galileo S-Band Contingency Mission, a mission based upon using the S-band LGA. This LGA mission includes major ground upgrades as well as inflight reprogramming of the Galileo spacecraft microprocessors to incorporate advance signal processing algorithms to boost the effective data rate. These onboard algorithms include advance error-correction coding, packetizing, and data compression schemes. A lossy image compression scheme known as the integer cosine transform (ICT) scheme [2] [3] was proposed, which is simple enough for spacecraft implementation. This scheme was extensively tested and was shown to provide good compression performance on images. It can also give a wide range of rate-distortion trade-offs for the image data, which accounts for over 70% of the total planned downlink data. In March 1993, the Galileo Project abandoned further attempts to free the HGA and adopted the LGA mission as the baseline.

ARC and JPL Collaboration. With ICT image compression algorithm baselined into the Galileo LGA mission, the evaluation and validation of this compression scheme with Galileo SSI principal investigators - in- the-loop is even more critical. The joint study conducted by ARC and JPL addressed this issue and resulted in validation of the ICT algorithm in terms of acceptability by the science user. The study incorporated representative images, anticipated noise and instrument signatures, quantization tables, expected compression ratios and most importantly, the science user community who evaluated and validated the expected compression scheme. Furthermore, the SSI principal investigators became more educated on the compression scheme and its effects on the visual quality of the Galilean images.

Ames' role was to develop the experimental design, implement the design, collect, and analyze the data from the subjects, and report findings and results. A pre-experiment survey of all members of the SSI was first conducted to collect preliminary information about the scientific interest of the expected imagery, what scientific questions are targeted for the images, how the questions are answered and what applications would be performed on the images. The survey results provided the basis for the PI-in-the-loop experiment. Subjective judgments and ratings were made by the scientists in a controlled environment at the Galileo SSI Compression Workshop held at NASA ARC. Ames collected, analyzed and reported the results to JPL.

JPL provided guidance to the ARC personnel and facilitated close communication with the SSI team members. JPL provided the ICT algorithm, library of representative images, quantization tables in support of the experiment.

ICT Algorithm. The ICT was chosen for the spacecraft because of its simplicity and performance. ICT can be thought of as an integer approximation of discrete cosine transform (DCT), which is regarded as one of the best transform techniques in image coding. The transform-based coding scheme consists of three stages: the data transform stage, the quantization stage, and the entropy coding stage. Both ICT and DCT are independent from source data statistics, and there are fast algorithms to perform ICT and DCT. Unlike DCT which requires floating-point or fixed-point operations, ICT requires only integer multiplications and additions, making it much simpler to implement than the

DCT. The elements in an ICT matrix are all integers, with sign and magnitude patterns that resemble those of the DCT matrix. Also the rows of the ICT matrix are orthogonal. The similarity of the ICT matrix to the DCT matrix, together with the orthogonality property of the ICT, guarantee that the ICT compression scheme performs almost as well as the DCT compression scheme, Joint Photographic Expert Group (JPEG).

METHODOLOGY

Basic Experimental Assumptions. We assumed that images can be grouped according to their visually based scientific features of interest and that experienced investigators having similar interests in these images have common requirements for acceptable visual fidelity. These assumptions permitted us to design an experiment around a reasonably small number of "representative" images as well as a manageable number of interested members of the SSI science team.

Experimental Design and Approach. The experimental design used to administer the variables of interest may be characterized as a 4 by 32 by 2 by 15 parametric design. The variables were:

q - Tables	4 tables
Quantization level	32 levels
Image type	2 (no noise; with noise)
Observers	15

Pair Comparison Method: Method of Paired Comparison was used [5]. Each observer was presented two compressed versions of the same image at a time side by side, varying only in their quantization level. They were not told anything about either image and only had to select which of the two possessed the highest overall quality to support them in conducting their visual examinations of that image. Then they rated each image on a scale from "1" to "5" where "1" represented a totally unacceptable scientifically-useless image, and "5" represented an image of the highest possible usefulness, value, or merit. A score of "3" was used as the threshold between acceptable and unacceptable for subsequent scoring purposes. No image pre-processing (contrast enhancement, stretching, etc.) were conducted on the images.

Method of Progressive Division: The Method of Progressive Division was used to quickly focus in and identify the optimal quantization level (q-level) for a given image and q-table, a group of observers were presented the same image and q-table with each person being presented a progressively smaller range of q-levels. The objective was to identify the quantization level(s) which separated an unacceptable from an acceptable rating. It will be recalled that a rating of "3" was considered as the threshold between an acceptable and an unacceptable image. Thus, images given a score that was higher or lower than "3" were used to determine when to decrease or increase the quantization levels, respectively, in subsequent testing. That acceptable half was presented to the next observer and bisected again, etc. This approach is based upon the (untested but reasonable) assumption that these observers possess a fairly consistent set of image evaluation criteria.

Observers: Fifteen people participated as subjects in the experiment. Six were SSI team members (representing six different institutions) while the remaining nine were participants at the workshop from another nine institutions. All possessed corrected or uncorrected 20:20 acuity and viewed the images on a high resolution SUN monitor.

Images Tested: Based upon meetings and telephone interviews with SSI team members at Ames and elsewhere we identified the following image classes of most interest to them. Images were selected for presentation for each of these seven classes from a larger image library provided by JPL. The experiment was conducted in a controlled environment at the SSI Compression Workshop held at Ames on July 22, 1993. Images were selected from each of the classes listed below, along with their respective noise-superimposed images.

Image Classes Studied

Solid surface with limb
Solid surface without limb
Solid surface with terminator
Gaseous surface without limb
Small bodies (e.g., asteroid)
Dark side phenomena/lightning
Rings

A total of fourteen separate images were studied in the experiment (cf. Table 1). Four represented the solid surface without limb category from Ganymede and Io. Three represented the solid surface with limb of Europa and Io, and another three represented a gaseous image without limb (all Jupiter). There was one image each representing a solid surface with terminator, small body (Gaspra), darkside phenomena (lightning), and rings (Saturn). All image files were cropped to fit side by side on the high resolution monitor and all but three were magnified x 2 in order to better demonstrate the effects of ICT compression. Four of the fourteen images were superimposed with noise frames.

Table 1 Image Details

Image Class Name	Body	File Name	Noise	Mag.	Q-tables		
					(1)	(2)	(3)
Solid with Limb	Europa	r.6.r		x 2	0	1	2
	Europa	r6.noise.r	x	x 2	0	1	2
	Io	r.9.r		x 2	0	1	2
Solid - No Limb	Ganymede	r.4.r		x 2	0	1	2
	Ganymede	rq538.g.r	x	x 2	0	1	2
	Io	sr7.raw.r		x 2	0	1	2
	Io	sr7.noise.r	x	x 2	0	1	2
Solid with Termin.	Callisto	r.1.r		x 2	0	1	2
Gaseous - No Limb	Jupiter	r.14.r		x 1	0	2	3
	Jupiter	r.15.r		x 1	0	2	3
	Jupiter	rq538.j4o.r	x	x 1	0	2	3
Small Bodies	Gaspra	rq538.gas.r		x 2	0	1	2
Darkside/Lightning	Earth	rq538.litn.r		x 2	0	1	2
Rings	Saturn	r.11.r		x 2	0	1	2

q -Table Selection: Four quantization (q) tables were developed for use in this study by A. B. Watson of Ames Research Center [8]. Each was designed to produce maximal ICT compression for different types of image characteristics, e.g., low contrast soft-boundary details, medium to high contrast high spatial frequency details.

RESULTS

Final Results Summary

Compression as a Function of Image Type: In general it may be said that the maximum ICT compression level(s) cannot be predicted apriori for a given image type and/or q-table. Nor are the perceptual response characteristics of observers understood well enough to predict whether unacceptable distortions of useful features with the digital image will be produced by the ICT algorithm at different q-levels. Visual ratings and associated commentary made by experienced observers/scientists are needed in order to determine how well a particular q-table and quantization level handles certain kinds of details. Nevertheless, the present data does provide some useful insights into the relative magnitude of acceptable compression ratios for different classes of images, noise types, quantization matrices, and levels presented.

The present data were grouped into a low, medium, and high acceptable image ICT compression ratio category. The low compression ratio group was selectively defined as ranging from no compression (1:1) to 4:1 and 8:1. The four images having superimposed noise all fell into this category regardless of which q-table was used.

There were three images in the medium acceptable compression ratio category (i.e., from 8:1 to 17:1), viz., r.1.r, r.4.r, and r.6.r. All three are solid surface images characterized by the presence of high spatial frequency details such as craters, linear structures, and other varied shapes of medium to high contrast.

The highest acceptable ICT compression ratio group was, on the basis of the present results, defined as higher than 35:1. Six images fell into this group. They are all relatively diverse from one another in image detail and deserve detailed commentary. Table 2 is a summary of acceptable image quality for each image type and q-table. The "Safe" range of compression values cited represent a more conservative (wider range of values) estimate of acceptable compression. These values take into account response variability. The "Likely" range represents our estimate of the actual range of compression ratios for each condition.

Influence of Radiation Noise: Four image types contained superimposed noise which would be expected to influence its visual appearance after compression. Three types of simulated radiation noise were studied. Two (Noise type B and D specified by JPL) consisted of random dots and short lines at random inclinations. Noise type C specified by JPL consisted of identical pairs of dots and short inclined lines separated by about 1/20th of the frame dimension. In three of these cases both a noise and non-noise version of the same image was quantified. It was found that radiation noise greatly reduces the ICT compression ratio that is judged as being acceptable to these observers. In the most extreme case found (r.15.r of the gaseous atmosphere of Jupiter vs. the same image with noise [rq538.j4o.r]) compression was reduced from 57:1 down to <3:1 (q-table 2) by the noise alone. In a less extreme case (r.6.r vs. r6.noise.r of Europa), compression of the same image was reduced from about 12:1 down to 5:1 (for q-table 0) due only to noise. In a third case involving a solid image without limb and high spatial detail (r.4.r vs. rq538.g.r of Ganymede) compression was reduced from about 10:1 down to 8:1 (q-table

Table 2
Summary of Acceptable Image Quality
Compression Results by Type of Image and q-Table

Image Type	file	Acceptance Criterion	q = 0	q = 1	q = 2	q = 3
Solid Surface with Limb	r.6.r	Safe Likely	8-12 8-12	9-15 9-15	4-12 8-12	----- -----
	r.9.r	Safe Likely	37-42 37-42	35-46 41-46	44-46 44-46	----- -----
	r6.noise.r	Safe Likely	1-5 4-5	< 2 < 2	< 3 < 3	----- -----
Solid Surface without Limb	r.4.r	Safe Likely	9-10 9-10	6-9 6-9	8-12 8-12	----- -----
	sr7.raw.r	Safe Likely	>38 >38	23-41 29-41	23-36 32-36	----- -----
	rq538.g.r	Safe Likely	4-8 4-8	< 3 < 3	< 4 < 4	----- -----
	sr7.noise.r	Safe Likely	1 1	< 2 < 2	< 2 < 2	----- -----
Solid Surface with Terminator	r.1.r	Safe Likely	11-17 11-17	12-15 12-15	11-18 11-18	----- -----
Gaseous Surface without Limb	r.14.r	Safe Likely	55-67 55-67	51-71 51-62	54-72 54-72	----- -----
	r.15.r	Safe Likely	36-53 36-53	----- -----	42-57 42-57	48-53 48-53
	rq538.j4o.r	Safe Likely	1 1	----- -----	< 3 < 3	6 6
Small Bodies	rq538.gas.r	Safe Likely	35-61 35-61	37-50 37-50	36-54 36-54	----- -----
	rq538.litn.r	Safe Likely	71-75 71-75	80-86 80-86	83-88 83-88	----- -----
Rings	r.11.r	Safe Likely	> 36 > 36	> 45 > 45	> 48 > 48	----- -----

0). Each q-table used produced slightly different results but of a comparable magnitude. In another image involving radiation noise (rq538.j4o.rof Jupiter) the q-table 0 image could not be compressed at all and still be acceptable. However, only two observers rated this image and neither responded to the instructions very seriously. Results for the q-table 2 and 3 yielded compression ratios of less than 3:1 and 6:1, respectively.

Compression as a Function of q-Table: By scanning vertically down Table 2 for each q-table one can quickly gain an understanding of the relative effect each q-table had on acceptable compression ratio by image. Q-table 0 yielded the highest acceptable ICT compression in only two (14%) of the fourteen images studied [viz., sr7.raw.r, and rq538.g.r]. Both are solid surface without limb. Q-table 1 yielded the highest acceptable ICT compression from 9:1 to 15:1 in only one (1%) of the fourteen images ([viz., r.6.r]. Q-table 2 yielded the highest acceptable compression in eight (57%) of the fourteen images studied.

GENERAL CONCLUSIONS

Radiation noise tends to reduce ICT compression acceptance ratings if high frequency information is desirable. Radiation noise also degrades low frequency information if the ICT compression used also eliminates high frequency information. The results showed that: (1) Acceptable compression ratios vary widely with the images; (2) Noisy images detract greatly from image acceptability and acceptable compression ratios; (3) Atmospheric images of Jupiter seem to have higher compression ratios of 4 to 5 times that of some satellite images.

DISCUSSION

It is clear that the impact of compression algorithms on images need to be studied further for specific science domains and specific principal investigators' scientific use for the images. Further, the ICT compression scheme is a block transform coding scheme. It performs lossy image compression, and it exhibits blockiness and checkerboard artifacts to different degree in the reconstructed image, depending on the image background and compression ratio. These block-oriented artifacts are caused by quantizing the transform coefficients of the ICT, and there are standard techniques in the literature to "remove" or "hide" these artifacts subjected to certain visual criteria. Most of the standard techniques assume no knowledge of the original image. The Galileo image compression scheme operates in a unique scenario where an addressable 96 pixel x 96 pixel area in an image can either be losslessly compressed or uncompressed (truth window). This area can provide valid statistics and boundary information to facilitate image reconstruction and artifacts removal. New and modified image restoration and enhancement techniques are now being developed to take advantage of the information provided by the truth window. New experimental procedures can be designed to evaluate the restoration and enhancement techniques by comparing the reconstructed images (with and without enhancements) with the original images. The PI-in-the-Loop approach can be a good approach to assess the validity of the compression techniques.

REFERENCES

- [1] Belton, M. J. S., K. P. Klaasen, M. C. Clary, J. L. Anderson, C. D. Anger, M. H. Carr, C. R. Chapman, M. E. Davies, R. Greeley, D. Anderson, L. K. Bolef, T. E. Townsend, R. Greenberg, J. W. Head III, G. Neukum, C. B. Pilcher, J. Veverka, P. J.

Gierasch, F. P. Fanale, A. P. Ingersoll, H. Masursky, D. Morrison, and J. B. Pollack. The Galileo Solid-State Imaging Experiment. Space Science Reviews, Vol. 60, No. 1-4, May 1992.

- [2] Cham, W., Development of integer cosine transforms by the principle of dyadic symmetry. IEEE Proceedings, Vol. 136, Pt. 1, No. 4, August 1989.
- [3] Cheung, K-M, and K. Tong, Proposed data compression schemes for the Galileo S-band contingency mission. Proceedings of the 1993 Space and Earth Science Data Compression Workshop, Pp. 99-109, Snowbird Conference Center, Snowbird, Utah, April 2, 1993, NASA CP 3191.
- [4] Ekroot, L., Preliminary ICT compression performance on Galileo images. Jet Propulsion Laboratory, Interoffice Memorandum 331-93.2-042, June 4, 1993
- [5] Guilford, J. P., The Method of Pair Comparisons, Psychometric Methods, McGraw-Hill, New York, 1954, Pp, 154-177.
- [6] Haskell, B. G., and R. Steele, Audio and video bit-rate reduction. Proc. of the IEEE, Vol. 69, No. 2, Pp. 252-262, February 1981.
- [7] Rabbani, M., and P. W. Jones, Digital Image Compression Techniques. SPIE Optical Engineering Press, Bellingham, Washington, Vol. TT7, Pp.108-111, 1991.
- [8] Watson, A., A. Alhumada, M. Young, Final Report: ICT Quantization Matrix Design for the Galileo S-Band Mission, Oct.1993.

IMAGE COMPRESSION SOFTWARE FOR THE SOHO LASCO AND EIT EXPERIMENTS

Mitchell R Grunes
ALLIED-SIGNAL TECHNICAL SERVICES
Russell A Howard
Karl Hoppel
Stephen A Mango
NAVAL RESEARCH LABORATORY
Dennis Wang
INTERFEROMETRICS

52-61
456
P-12

ABSTRACT

This paper describes the lossless and lossy image compression algorithms to be used on board the Solar Heliospheric Observatory in conjunction with the Large Angle Spectrometric Coronagraph and Extreme Ultraviolet Imaging Telescope experiments. It also shows preliminary results obtained using similar prior imagery and discusses the lossy compression artifacts which will result. This paper is in part intended for the use of SOHO investigators who need to understand the results of SOHO compression in order to better allocate the transmission bits which they have been allocated.

INTRODUCTION

The Solar and Heliospheric Observatory (SOHO) is currently scheduled for a July 1995 launch into a lunar L1 orbit. The software described will compress images from the Large Angle Spectrometric Coronagraph (LASCO) (a wide-field white light and spectrometric coronagraph) and the Extreme Ultraviolet Imaging Telescope (EIT) experiments. LASCO will image the solar corona from about 1.1 to 30 solar radii, and has a built in spectrometer to measure, point-by-point, plasma temperature, density, bulk and turbulent velocities, and the direction of the magnetic field.

The transmission bandwidth (5200 bits/sec) is insufficient to transmit the desired imagery. In order to resolve this problem, our software implements two image compression algorithms:

1. A lossless image compression algorithm.
2. A lossy image compression algorithm, expected to be used for most of the imagery. In most cases investigators are expected to select an output of about 1.6 bits/pixel (bpp), a compression factor of 10 from the input 16 bit format. This will allow transmission of about 240 images/day, plus some other overhead and small transient images.

The code is mostly written in the C programming language. It will run on a Sandia SA3300 CPU, a rather slow (about 1 MIPS) radiation hardened space qualified processor which was designed to emulate a National Semiconductor 32C016 Series CPU.

The relatively slow data rate allows us to use compression algorithms which are of higher quality on the solar test imagery than published standards such as JPEG, in spite of the hardware limitations of the target computer. This was accomplished at the cost of increased complexity and processing load. However, these are acceptable for our application because:

1. The data will be gathered at substantial cost.

2. As in many space applications, the allotted transmission bandwidth is the major limiting factor on the transmitted spatial and radiometric resolution, and on the frequency with which images can be transmitted. This is because transmission bandwidth translates directly to power and storage requirements, and thus to the weight and cost of the satellite.

As in many space applications, the imagery will be reconstructed (decompressed) by a work station on the ground with much more computing power than the compressing computer.

Some comparisons with the independent JPEG algorithm will also be given.

This statistics that appear in this paper are somewhat preliminary. The final paper may use somewhat different algorithms which may produce better results. In particular, several changes to our algorithms will be investigated in order to insure that the result is as close to the optimal as is practical within the constraints of the target processor. For lossless compression this might include the use of a non-integral number of bits to code the least significant fraction of the split coder, or the use of adjusted binary codes after the style of Golomb. It is not clear at this time what this might include for lossy coding.

LOSSLESS COMPRESSION ALGORITHM

The method described in

Rice, "Some Practical Noiseless Coding Techniques, Part III, Module PSI14,K+", JPL Publication 91-3, 11/91

served as a starting point for the development of the lossless compression algorithm because:

1. It requires relatively little code or time to implement.
2. Very few bits are needed to provide small block size adaptivity. This is important because there is expected to be a great deal of difference in brightness and texture between different parts of the image, and because CCD array sensors develop small area defects.

Various changes were made to that algorithm. In brief:

1. Different choice of block size, and the use of bi-level two dimensional blocks.
2. More adaptive classes.
3. Triplet coding was not implemented because it is anticipated that the 14 to 16 bit images will be statistically random in the lower few bits.
4. A somewhat improved prediction algorithm.
5. A somewhat more complex coding technique was used to keep down the number of bits used for adaptivity.

An optimal DPCM technique was also investigated. The weights were determined by a least squares fit. This produced predictions which were then input into the modified Rice algorithm. This improved the compression factor by only 5% for the 13 bit eclipse image. The improvement will probably be even smaller for the 14 and 16 bit imagery that the software will be applied to. Hence it was decided that it was not worth performing least squares processing to determine optimal weights.

Small scale adaptivity outweighs the advantages of more sophisticated entropy coding. For example, it significantly out-performs pure Huffman coding techniques on sample images similar to those expected from SOHO. In fact it performs somewhat better than would appear to be

possible on the basis of *whole-image* "entropy" measured in terms of the frequency of original pixel values, or in terms of the differences from predicted pixels. Note, however, that some methods, such as lossless JPEG, do produce better results for many 8 bit images. It is quite possible that a better algorithm may be used in the final software.

LOSSY COMPRESSION ALGORITHM

The ADCT (Adaptive Discrete Cosine Transform) method described in

Chen and Smith, "Adaptive Coding of Monochrome and Color Images", IEEE vol 25 #11, Nov 1977, pp. 1285-1292

served as a starting point for the development of the lossy compression algorithm because:

1. It is a method with which NRL Code 7230 has a great deal of experience. We have implemented that algorithm (somewhat differently) in a software package which has been used operationally for some time by various U.S. government agencies.
2. It is a fully adaptive ADCT, which chooses the number of bits used to specify each DCT transform coefficient within each class of block according to its activity. No a priori statistics are required.
3. Max-Lloyd Gaussian quantization is used in the frequency domain, which performs much better than uniform quantizers.
4. One may specify a definite compression factor can be specified over a large, fairly continuous range.
5. It is not especially fast or simple, but it is certainly faster than known high quality fractal and vector quantization algorithms.
6. It remains one of the very best image compression methods yet developed, performing better than many of the more recently published algorithms.

Various changes were made to that algorithm, some of which improve upon our earlier work. In brief:

1. A different block size was chosen, to improve quality, and to mesh better with other intended spacecraft processing.
2. More block classes (up to 16, depending on image size) and a somewhat different method of separating classes (a compromise between block variance and maximum coefficient scaling) is used. These changes were done in order to largely eliminate the discontinuities in brightness and texture that occurred across block boundaries, at the price of somewhat larger RMS pixel errors.
3. The quantization tables are normalized somewhat differently.
4. Very low intensity coefficients are randomized to prevent systematic quantization errors leading to bright or dark spot artifacts.
5. Several details not specified by Chen and Smith were provided by us, such as:
 - a. The bit allocation table is sent efficiently, employing run length encoding of alternate direction diagonals.
 - b. The coefficients are scaled so as to emphasize the most visible features.

The modified algorithm produces surprisingly good results. In particular, the existence and position of edges remains accurate up to fairly high compression factors (but some blurring occurs, there are echoes and shifts in the radiometric centers of isolated bright points, and there

are some discontinuities at block boundaries). Preliminary work using full search vector quantizations did not yield as good results. Wavelet transform methods might produce more continuous results across block edges, but that did not generally appear to be a problem for the sample images at the desired compression factors.

As a test, the eclipse image was compressed and reconstructed using the lossy algorithm. The difference image was then compressed using lossless compression. The total number of bits used was about the same as to code the image using lossless compression alone. Therefore the lossless and lossy algorithms store about the same amount of information per bit.

APPROPRIATE IMAGERY AND COMPRESSION FACTORS

The software was written to apply to 2 dimensional continuous tone monochrome still imagery, with up to 16 bits/pixel. A number of arbitrary factors in the design were decided on the basis of the solar test imagery.

Both the lossless and lossy compression algorithms perform best with images which are somewhat smooth. For example, they will not perform very well with images that have been digitized in a small number of bits or quantized at a small number of levels, such as dithered images, nor with extremely noisy images, such as one-look SAR.

Both the lossless and lossy compression algorithms perform sub-optimally on images which are so smooth that a significant fraction of pixels are perfectly predictable from their neighbors; the 14-16 bit quantization of our input data will probably contain noise or small scale features in the lower few bits.

The lossy algorithm performs sub-optimally on isolated bright and dark spots or lines, although edges between two regions of differing brightness are represented fairly well. In addition, images containing features with a very wide dynamic range may tend to distort small features with low contrast levels, and some noise is introduced into very low contrast areas. For example, images consisting of many stars or spectral lines would be inappropriate.

If lossless compression is applied to inappropriate images, substantially more bits will be used than are needed. Lossy compression of inappropriate images will blur features, shift the radiometric centers of isolated bright and dark spots, and introduce shape distortions or lose small and subtle features. It may also introduce discontinuities in brightness and texture at block boundaries.

For this project the lossy compression software was intended to be applied at a compression factor of 10 to 15 relative to 16 bit/pixel input, yielding 1.07 to 1.6 output bits/pixel. The algorithm can produce adequate results at somewhat smaller compression factors, and it could theoretically be applied at compression factors up to several hundred. In practice, the inefficiencies due to packet format and small block size make our implementation inappropriate above a compression factor of about 20.

Applying lossy compression with excessive compression factors yields problems similar to applying it to inappropriate images.

DEFINITION OF TERMS

There are a number of terms that we use in evaluating the performance of our software. These terms are defined in many different ways by different researchers.

Compression Factor relative to the 16 bit/pixel input format:

$$CF_{16} = \frac{\text{Bits in original image (at 16 bits/pixel) with no overhead}}{\text{Bits in compressed image with overhead including packets}} \quad (1)$$

RMS Error

$$RMS\ Error = \sqrt{\text{Mean Square (original image - reconstructed image)}}$$

Note that RMS error is very close to standard deviation for both our technique and the independent JPEG algorithm, because systematic bias is negligible in both cases.

Normalized Mean Square Error:

$$NMSE = \frac{\text{Mean Square (original image - reconstructed image)}}{\text{Mean Square (original image)}} \quad (3)$$

Other definitions of NMSE, in which the mean square pixel value is replaced by the maximum or maximum possible value, are quite common. Errors shall be reported both for pixels and for gradients (first differences, taken along both image directions). The former is scientifically meaningful because plasma brightness can be related to total electron content, the latter because feature detection and recognition depends on detection of edges and texture.

Throughout this paper we have omitted the approximately .0625 bits/pixel to be expected in compression packet overhead, as well as the overhead to be used for other types of packets and transmitted information.

TEST IMAGES

We use 5 test images. We shall also test with parts of images masked out. Masking will sometimes be used in the spacecraft to omit parts of the image covered by the occulter. (Occulter is used to eliminate very bright light which would otherwise wash out the desired imagery.) Masking is a very simple form of additional compression, which eliminates the bits needed to code the masked out features.

Image Name	Pixel Columns*Rows*Bits	Actual Source Instrument	Similar to LASCO/EIT telescope
Eclipse	512*512*13	Ground Photograph	C1
Same, masked		" "	C1
Same, masked ₂		" "	C3
Vidicon	512*512*11	Solar Max	C2
Same, masked		" "	C2
Helio	512*1024*13	HRTS Spectroheliograph	EIT
H _α	1024*1024*14	HRTS H _α	EIT, but lower contrast
Lenna	512*512*8	Human Photograph	None

The HRTS images were summed in 2*2 pixel blocks to reduce the data to the approximate resolution of EIT. Note that the Lenna (sometimes Lena) image has been included simply because it is probably the most commonly used test image in the image processing field. No importance was given to getting good results with Lenna.

All of the test images except Lenna are shown in the figures.

LOSSLESS COMPRESSION RESULTS

Results are first listed for the original test image. 16 bit rescaled values are also very pessimistically estimated by assuming that the additional bits are random. Real imagery should perform better.

Image	CF ₁₆	bits/pixel	CF ₁₆ , rescaled	bits/pixel, rescaled
Eclipse	2.24	7.13	1.58	10.13
Same, masked	2.35	6.80	1.63	9.8
Same, masked ₂	2.64	6.06	1.77	9.06
Vidicon	3.96	4.04	1.77	9.04
Same, masked	4.64	3.45	1.89	8.45
Helio	1.63	9.80	1.25	12.8
H _α	1.77	9.04	1.45	11.04
Lenna	3.37	4.75	1.25	12.75

LOSSY COMPRESSION RESULTS

Lossy compression, by definition, involves the loss of information. The following table represents the results of compressing the test images to a nominal 1.6 bits/pixel:

Image	Pixel Error				Gradient Error			
	SOHO		JPEG		SOHO		JPEG	
	RMS	NMSE	RMS	NMSE	RMS	NMSE	RMS	NMSE
Eclipse	10.9	1.1E-5	19.2	3.3E-5	13.8	.077	25.6	.263
Same, masked	10.2	9.2E-6	n/a	n/a	13.1	.069	n/a	n/a
Same, masked ₂	9.3	9.3E-6	n/a	n/a	11.8	.080	n/a	n/a
Vidicon	1.7	8.6E-5	3.7	4.1E-4	2.1	.038	5.3	.241
Same, masked	1.5	5.7E-5	n/a	n/a	1.9	.025	n/a	n/a
Helio	100.5	5.2E-3	114.5	6.8E-3	135.0	.026	173.6	.425
H _α	58.2	1.4E-4	70.9	2.1E-4	85.0	.123	104.1	.185
Lenna	3.5	9.8E-4	3.24	8.3E-4	4.6	.156	5.0	.183

It was not practical to provide JPEG results for the masked images, because the independent JPEG code, as supplied did not implement masks.

Pixel errors are better than those from JPEG, partly because the independent JPEG software was designed to handle 8 bit imagery, so our imagery was scaled to fit. (With real 8 bit test imagery, the results were mixed.) The exception is Lenna, where JPEG does noticeably better. It is our belief that the very extensive use of Lenna, together with RMS error or NMSE, in the compression literature, caused the JPEG and independent JPEG algorithms to be somewhat biased to produce good results with that image.

Gradient errors are uniformly better than those from JPEG, partly for the same reasons, but partly because gradient errors are rarely looked at, so that they probably did not much influence JPEG design.

The figures show that there is very little visual loss. For all images the major apparent change is a blurring of isolated bright and dark points. There is also some noticeable blurring of edges, and there is a modification and introduction of some noise into low contrast features. Overall, however, the compression quality is excellent.

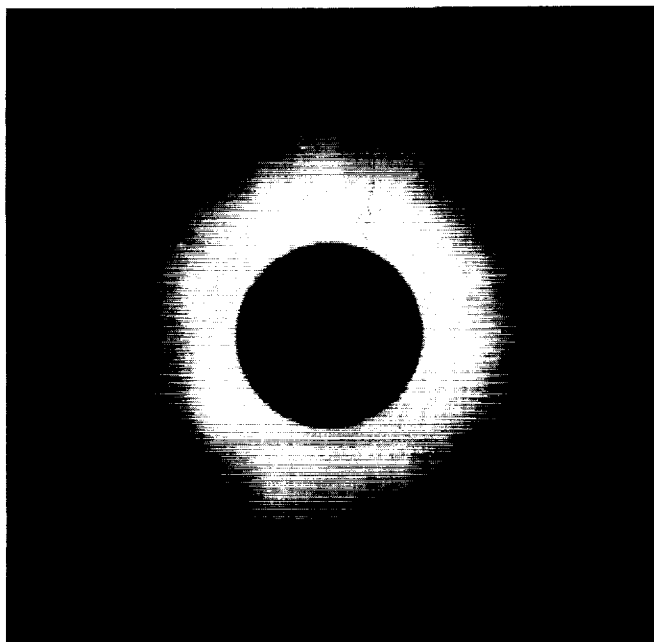


Fig. 1A Original Eclipse image, 512*512 pixels

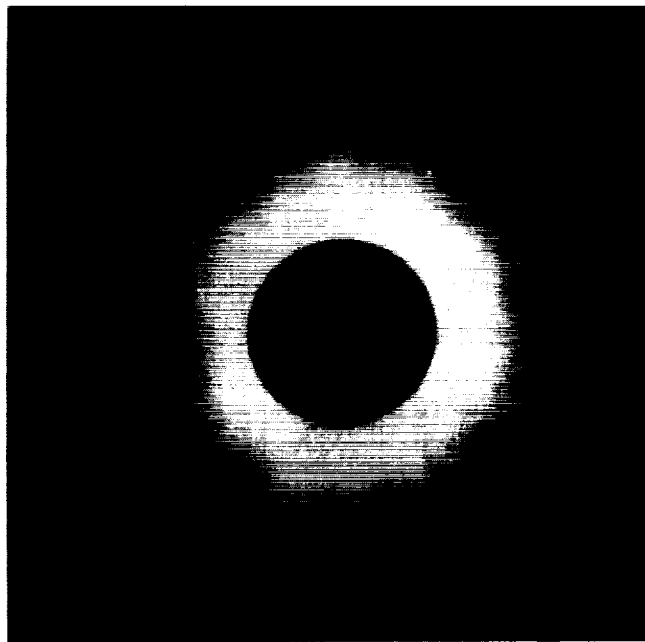


Fig. 1B SOHO compression to about 1.6 bpp

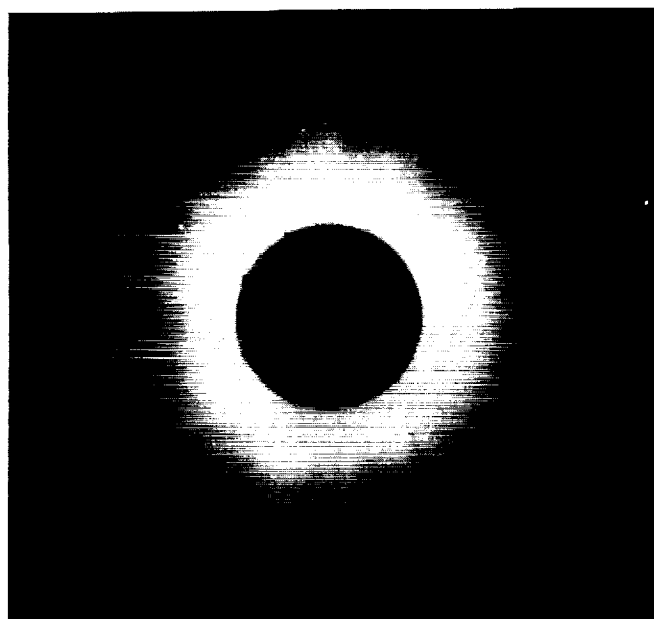


Fig. 1C Same, with C1-like mask



Fig. 1D Same, with C3-like mask



Fig. 2A Stretched 64*64 pixel section of 1A

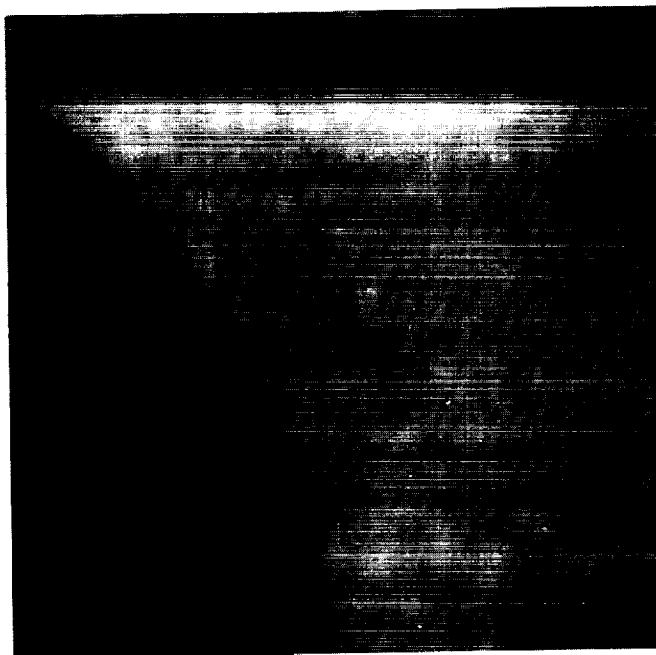


Fig. 2B Same for 1B

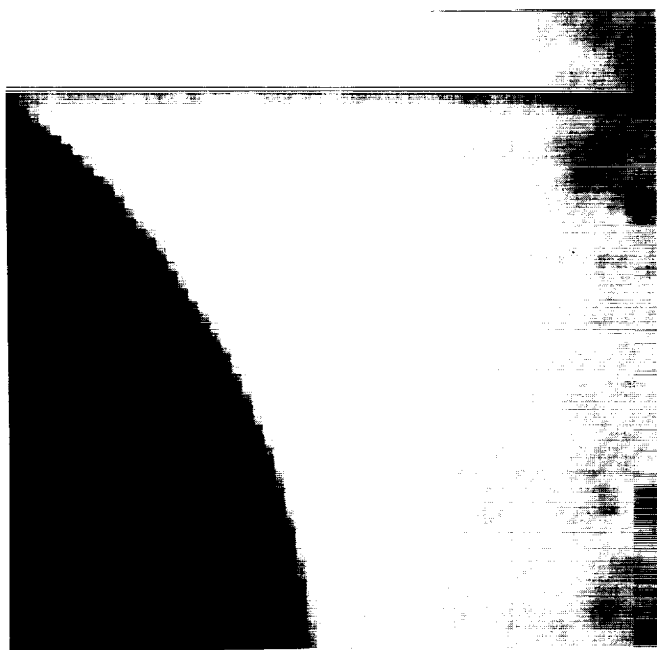


Fig. 2C Same for JPEG compression

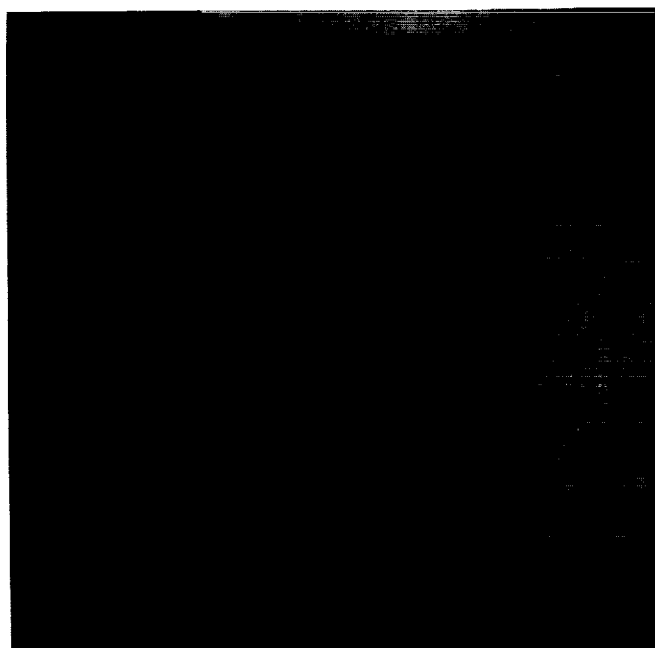


Fig. 2D Same for 1D

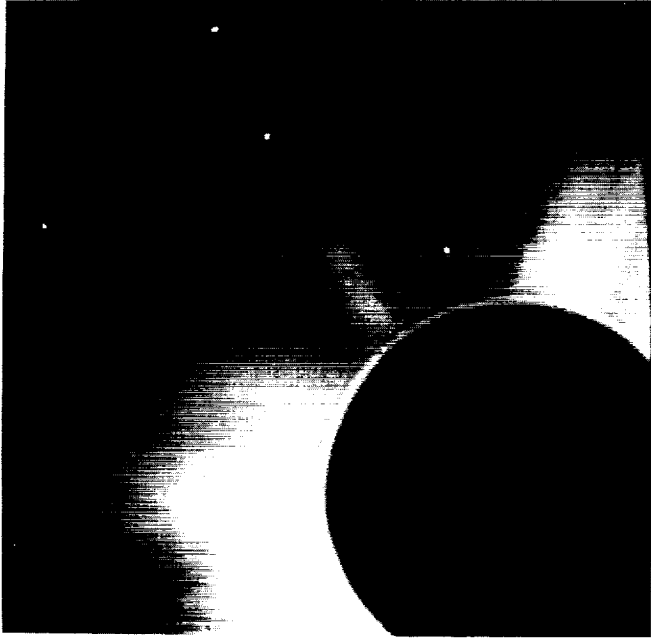


Fig. 3A Original Vidicon image, 512*512 pixels

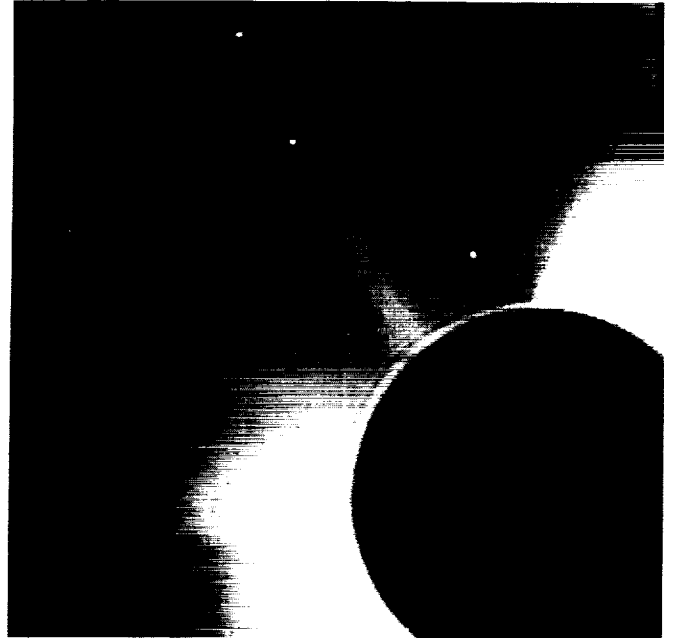


Fig. 3B SOHO compression to about 1.6 bpp

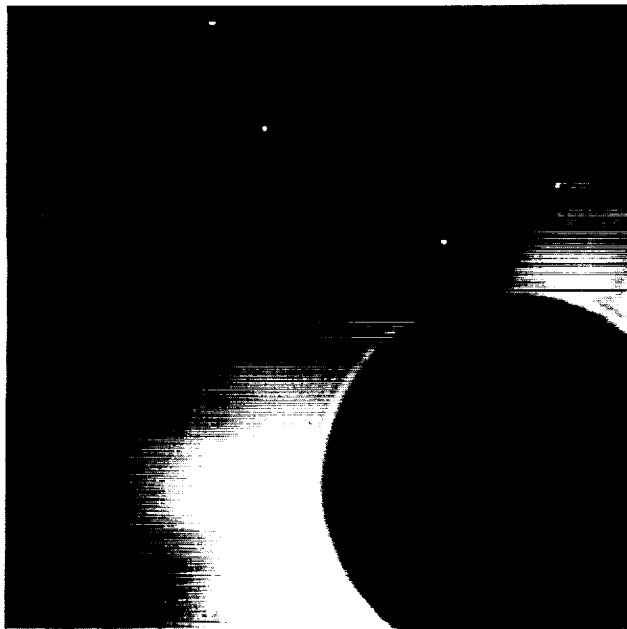


Fig. 3C Same with mask

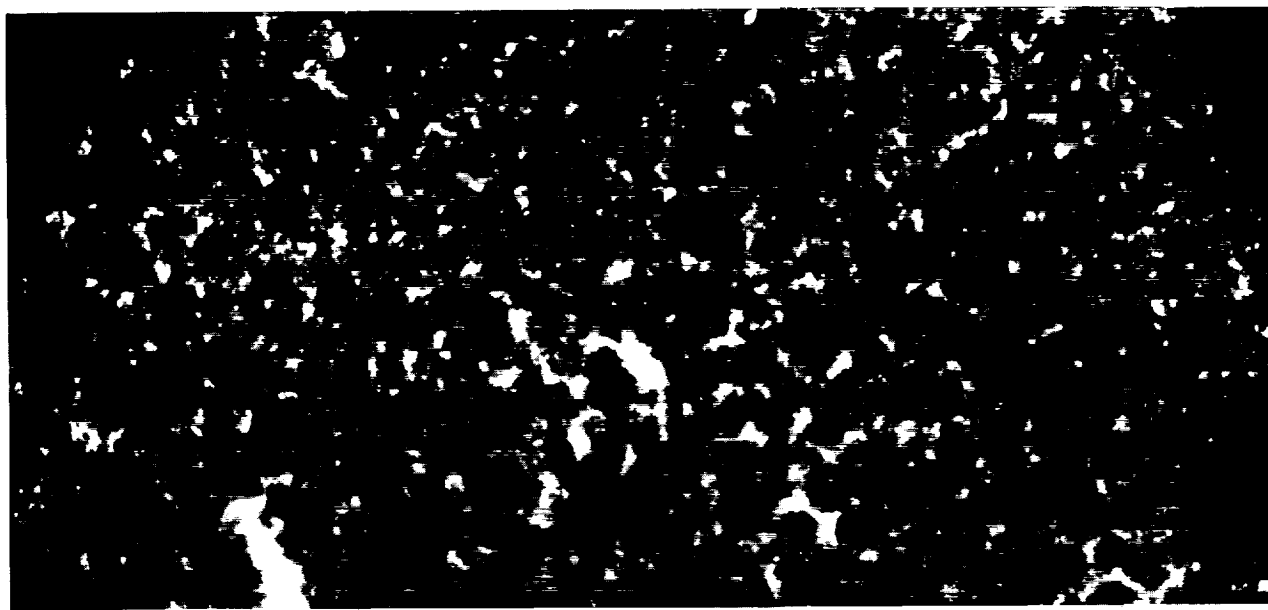


Fig. 4A Original Helio image, 512*1024 pixels

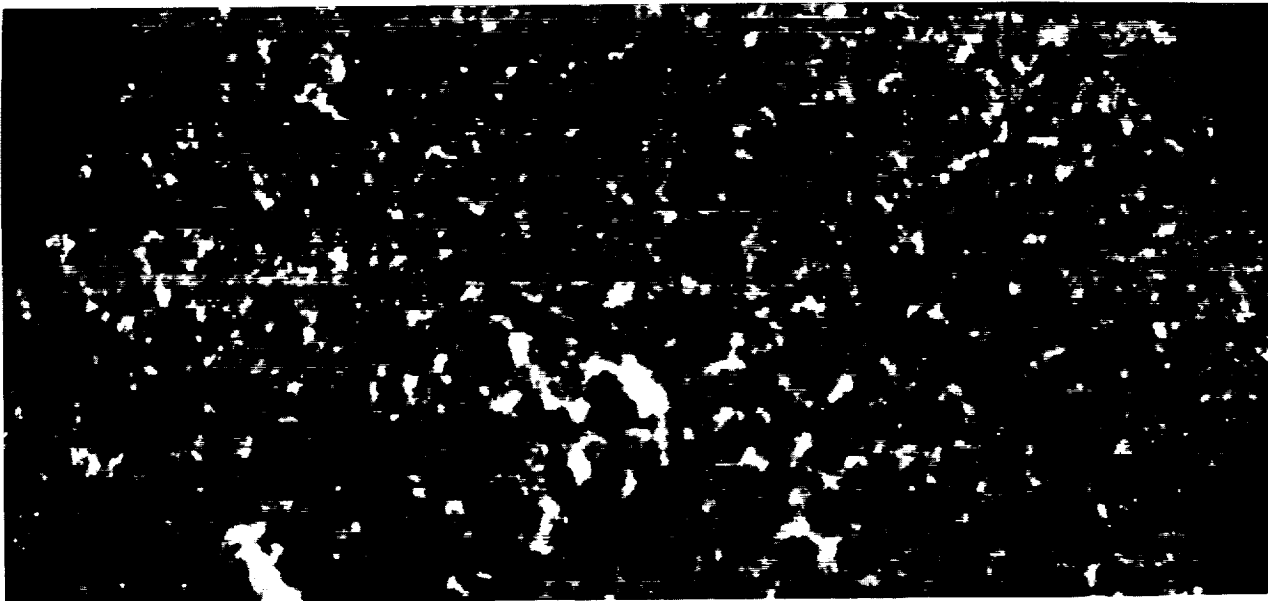


Fig. 4B SOHO compression to about 1.6 bpp

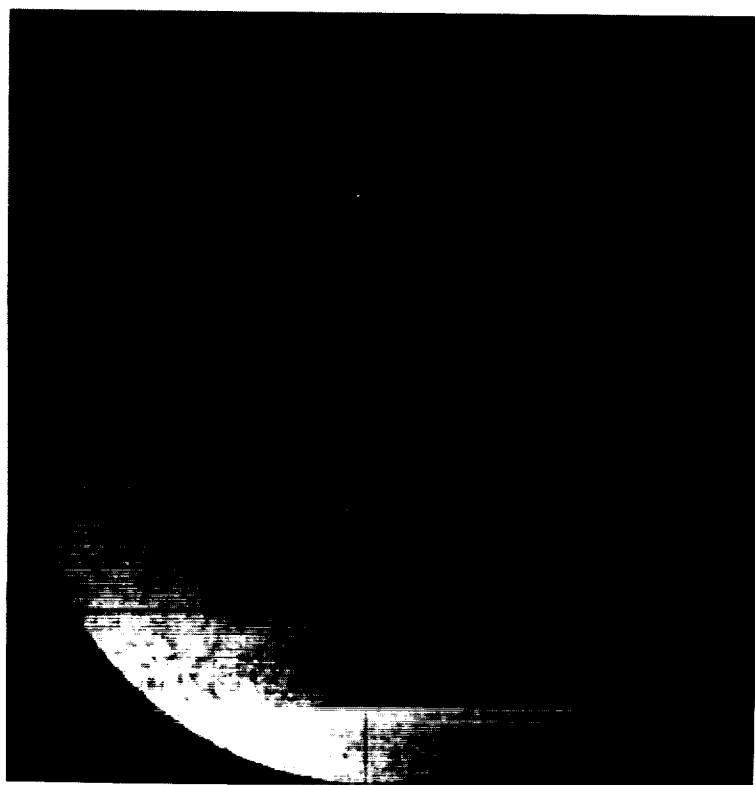


Fig. 5A Original H-alpha image, 1024*1024 pixels

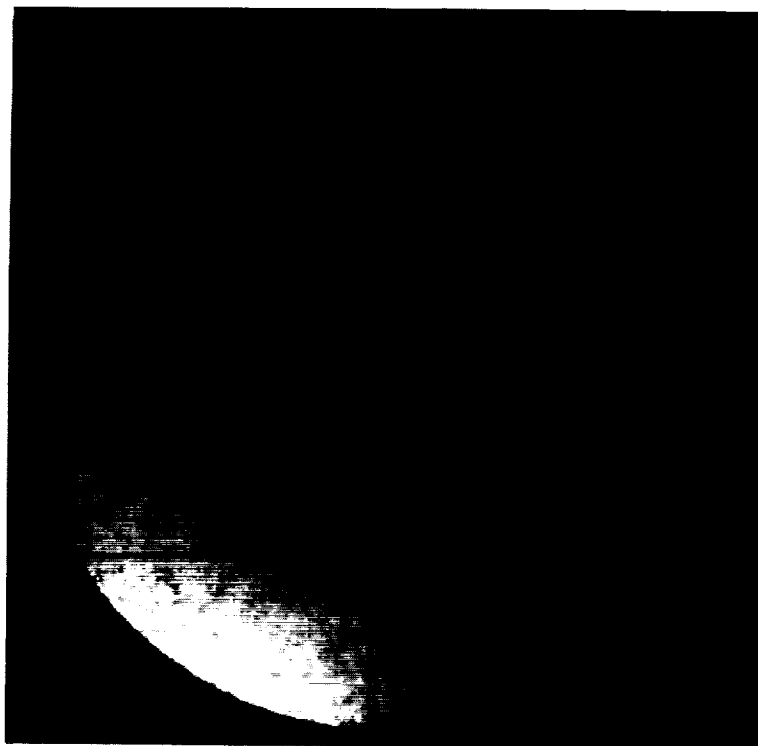


Fig. 5B SOHO compression to about 1.6 bpp

Comparison of the Lossy Image Data Compressions for the MESUR Pathfinder and for the Huygens Titan Probe

P. Rüffer ¹, F. Rabe ¹, F. Gliem ¹, H.-U. Keller ²

¹ Technische Universität Braunschweig, Postfach 3329, D-38023 Braunschweig

² Max-Planck-Inst. f. Aeronomie, Postfach 20, D-37191 Katlenburg-Lindau

53-61

457

P. 8

Abstract : The commercial JPEG standard complies well with the specific requirements of exploratory space missions. Therefore, JPEG has been chosen to be the baseline for a series of spaceborne image data compressions (e.g. MARS94-HRSC, -WAOSS, HUYGENS-DISR, MESUR-IMP). One S/W-implementation (IMP) and one H/W-implementation (DISR) of image data compression are presented. Details of the modifications applied to standard JPEG are outlined. Finally a performance comparison of the two implementations is given.

1 Introduction

This paper introduces two lossy image data compressions designed for exploratory space missions. Both compressions represent task oriented modifications of the Joint Photographic Expert Group (JPEG) standard for still image data compression [1]. Accordingly, both are based on Discrete Cosine Transform (DCT).

For the NASA/ESA Cassini/Huygens Descent Imager Spectral Radiometer (DISR)¹ [2] the mission profile required the development of a dedicated compression hardware. Apparently, both the mission profile of the NASA Imager for MESUR Pathfinder (IMP)² [3] and the availability of a RISC central board computer supported a completely software oriented implementation. The modifications of the JPEG scheme can be categorized as :

- (a) simplifications for H/W savings (DISR)
- (b) improved data dropout robustness
- (c) adaption of compression algorithms to the actual scene

¹Principle Investigator : M.G. Tomasko, Univ. of Arizona

²Principle Investigator : P. Smith, Univ. of Arizona

2 JPEG baseline scheme

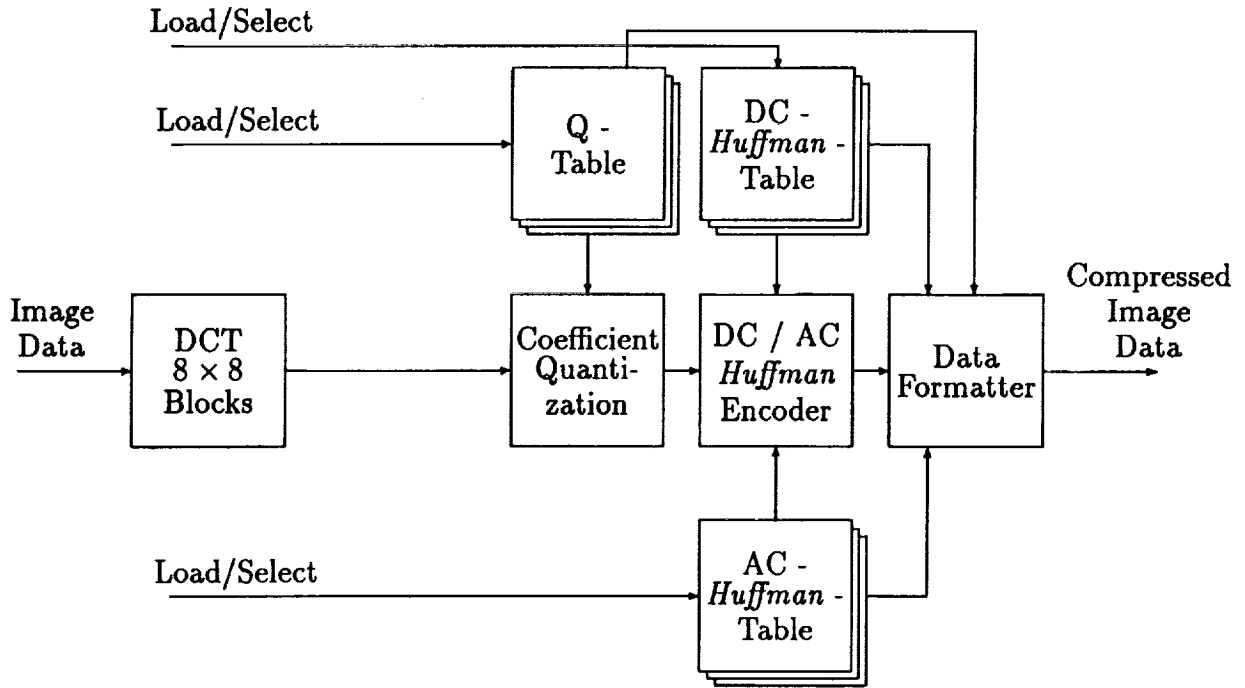


Figure 1: Data/control flow of JPEG sequential DCT baseline scheme

The JPEG standard describes a collection of image compression tools from which a subset can be selected to satisfy application specific requirements. JPEG offers four modes of operation (1) Sequential DCT, (2) Progressive DCT, (3) Sequential lossless and (4) Hierarchical mode. Sequential DCT (1) is well established and is implemented within numerous H/W- and S/W-applications. Therefore, the "baseline system" option of sequential DCT was selected as the compression scheme for IMP and DISR.

The sequential DCT mode consists of a "baseline system" and an "extended baseline system". Contrary to the "extended baseline system" the "baseline system" represents a minimum of coding flexibility, defined by the capability of the decoder. This scheme is splitted into a sequence of DCT-operation, coefficient quantization and *Huffman* coding (see Figure 1). Finally a data formatter organizes the compressed data.

DCT based transform coding is well suited for compression of pixel data with high correlation between adjacent pixels. Application of the DCT to a $N_1 \times N_2$ array of pixel intensity values (image domain) maps these values into a $N_1 \times N_2$ array of coefficients (frequency domain). Because of the DCT energy packing nature most of the image energy now is concentrated into a small number of neighbouring and highly decorrelated coefficients. The residual majority of coefficients represents a small fraction of image energy only.

Moderate savings of computing time (DCT operation) and limitation of error propagation are the rationals for the subdivision of the image array into nonoverlapping blocks each of

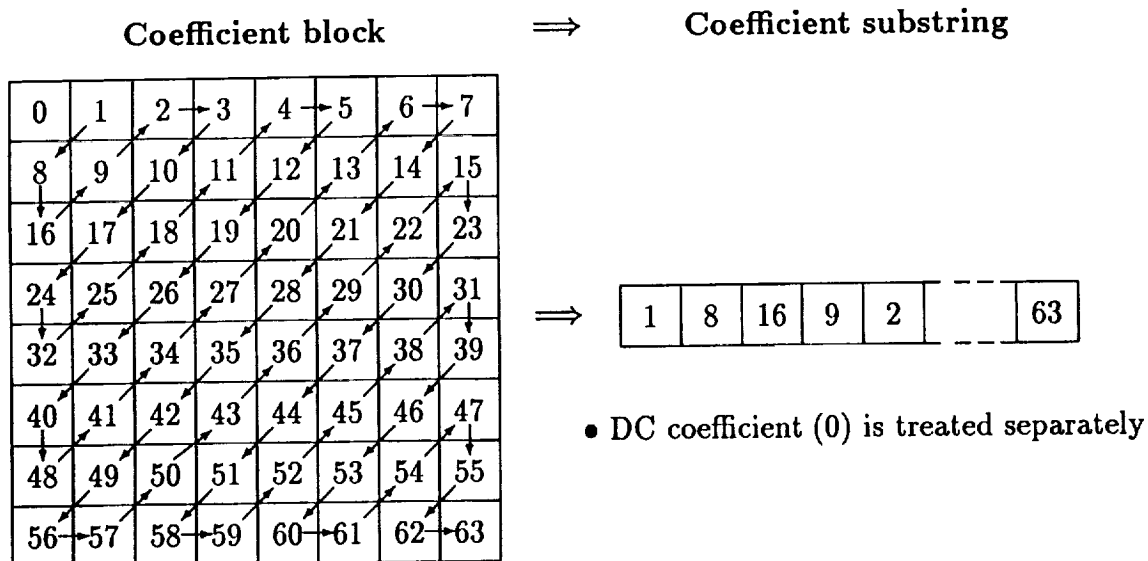


Figure 2: Rearranged coefficient block

size $M \times M$ pixels. However, signal to noise ratio degrades with decreasing block size. $M = 8$ and $M = 16$ provide a reasonable compromise between these contradictory constraints.

In order to increase the coder efficiency the coefficients of the two-dimensional array are rearranged in zigzags to a one-dimensional string representation (Figure 2) [4]. The distance between coefficient locale and the upper left corner reflects the spatial frequency. The coefficient values have the tendency to decrease with increasing spatial frequency. Coefficients with values below a coefficient dependent low bound are set to zero in the case of quantization. Therefore zigzag rearrangement increases the length of "zero" sequences.

Data compression is achieved by

1. coefficient quantization, which reduces the accuracy and therefore the number of bits per coefficient (lossy operation)
2. coding which optimizes (reduces) the average word length of coefficient representation (lossless operation)

The baseline system operation of coefficient quantization is based on the model of an uniform quantizer. It uses an individual quantization step width for each coefficient of the substring and for the DC value.

Quantization values are set individually using performance criteria such as human visibility or any kinds of image signal qualities. They are stored using a zigzag arranged quantization table (Q - Table). JPEG offers the selection of one out of four possible Q-Tables. The selection is fixed for the complete image. Compression amount is user controlled by a factor called *quality level*. Depending on this factor the quantization values of the actual Q-Table are rescaled before the quantization starts.

The baseline system distinguishes the coding of the single DC-coefficient and the $M^2 - 1$ AC-coefficients. While there is only one DC-coefficient for each coefficient block it is sufficient to code the DC magnitude only. Accordingly coding of the AC-coefficients involves both, coding of the coefficient magnitude as well as coding of the coefficient position.

3 Requirements derived from mission profiles

	mission target	experiment operation time	averaged data rate	total amount of data	image rate	implemen- tation
DISR	Titan	$\approx 2.5h$	450 bps	4 Mbit/mission	10/s	H/W
IMP	Mars	30 d - 1 a	600 bps	50 Mbit/d	0.2/min	S/W

Table 1: Mission profiles

The major aspects of the mission profiles are summarized in Table 1. IMP will be launched in 1996 and will land on Mars in 1997. During a 30 days primary and a second operation which is extended to one year IMP will take different kinds of images (single images, panorama) and will monitor the rover operation. Analysis of preceding images will be used to define both the best suited imaging mode and compression mode. Requirements for the IMP image data compression are

- (a) a 256×256 image has to be compressed within 5 minutes
- (b) automatic operation, but human interaction
- (c) self adaption to spatially varying image statistics, target compression factor selectable, image quality adjustable
- (d) compliance with RISC board computer capability

Due to the moderate image rate (see (a)) no dedicated H/W is needed. Unfortunately, this comfortable and flexible situation is not applicable to the Huygens Camera.

Cassini with its daughter probe Huygens will be launched in 1997 and will arrive at Saturn moon Titan in 2006. After release by the orbiter the probe will descend through Titan's atmosphere down to its surface within approximately 2.5 hours. Only during this descent DISR will take, preprocess, compress and transfer images. Due to this mission profile the image data compression concept for DISR has to comply the following requirements :

- (a) a 256×256 pixel image has to be compressed in less than 0.1 s
- (b) completely automatic operation, human interaction via telecommand is impractical because of signal propagation time (70 min. one way, 150 min. operation time)
- (c) self adaption to spatially varying image, fixed set of target compression factors

- (d) compliance with environmental requirements as board area (225cm^2), mass (210g), peak power (0.6W) and averaged power consumption (0.4W @image frequency = 10 images/s)

Driven by these tough requirements a dedicated hardware solution has been implemented for DISR.

4 IMP image data compression

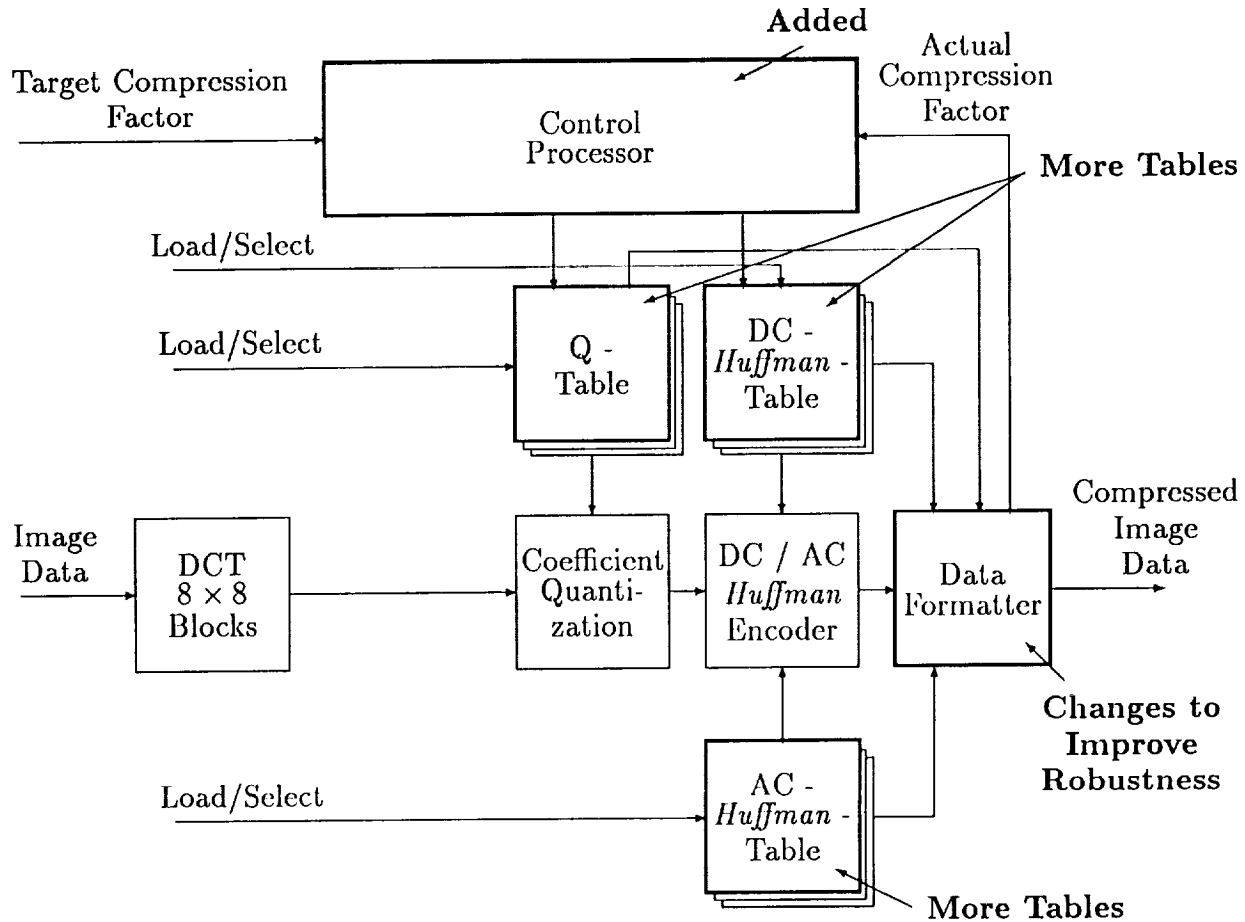


Figure 3: Data/control flow of IMP image data compression scheme

The IMP compression is a pure S/W solution based on the JPEG baseline system. According to mission specific requirements baseline system algorithm has been stripped down to serve only monochrome images. Further all not applicable parameters have been removed from the output data format.

Generally, entropy/redundancy reduction increases the tendency of error propagation in case of telemetry dropouts. To cope with this serious problem the following modifications have been implemented :

(a) **JPEG** : *Q*- table loadable, table contents are included in each compressed image data set

IMP : 16 loadable *Q*- tables, selectable by telecommand, multiple table references instead of full table contents are included in each compressed image data set.

(b) **JPEG** : *Huffman* table individually generated for each image is included in each compressed image data set.

IMP : 16 loadable *Huffman* tables, selectable by telecommand or automatically for highest compression ratio. Multiple table referencing as (a)

(c) **IMP** : in order to restrict error propagation to block boundaries a specific image position identifier has been added

Further, an optional feedback path has been implemented for the iterative adjustment of the compression factor to a given target value.

Arithmetic coding as proposed by JPEG improves coding efficiency. Error robustness requires additional synchronization means, which degrades the performance of arithmetic coding. Whether a reasonable balance does exist, shall be investigated by simulations being in progress.

5 DISR image data compressor

As stated before the DISR task is characterized by a rather high image rate of 10 images per second. Phase A/B studies have shown that the handling of this rate requires the design of a specific H/W processor[5]. This design was based on the Thomson DCT Processor STV3200, which provides sufficient radiation hardness.

Again, the processing scheme is rather similar to JPEG. Modifications are mainly directed to hardware savings. The most prominent modifications are :

(a) **JPEG** : 8×8 blocks

DISR : 16×16 blocks, provides a slightly improved compression ratio at the expense of a slightly degraded error robustness

(b) **JPEG** : Individual *Q*-value for each coefficient of a block

DISR : Coefficient quantization is subdivided into coefficient qualification by threshold (*th*) and quantization of the remaining coefficients. Coefficients are quantized using one unique (adjustable on image level) *Q*-value. Deletion map provides efficient coding of deleted coefficients.

(c) **JPEG** : *Huffman* coding

DISR : Run length coding

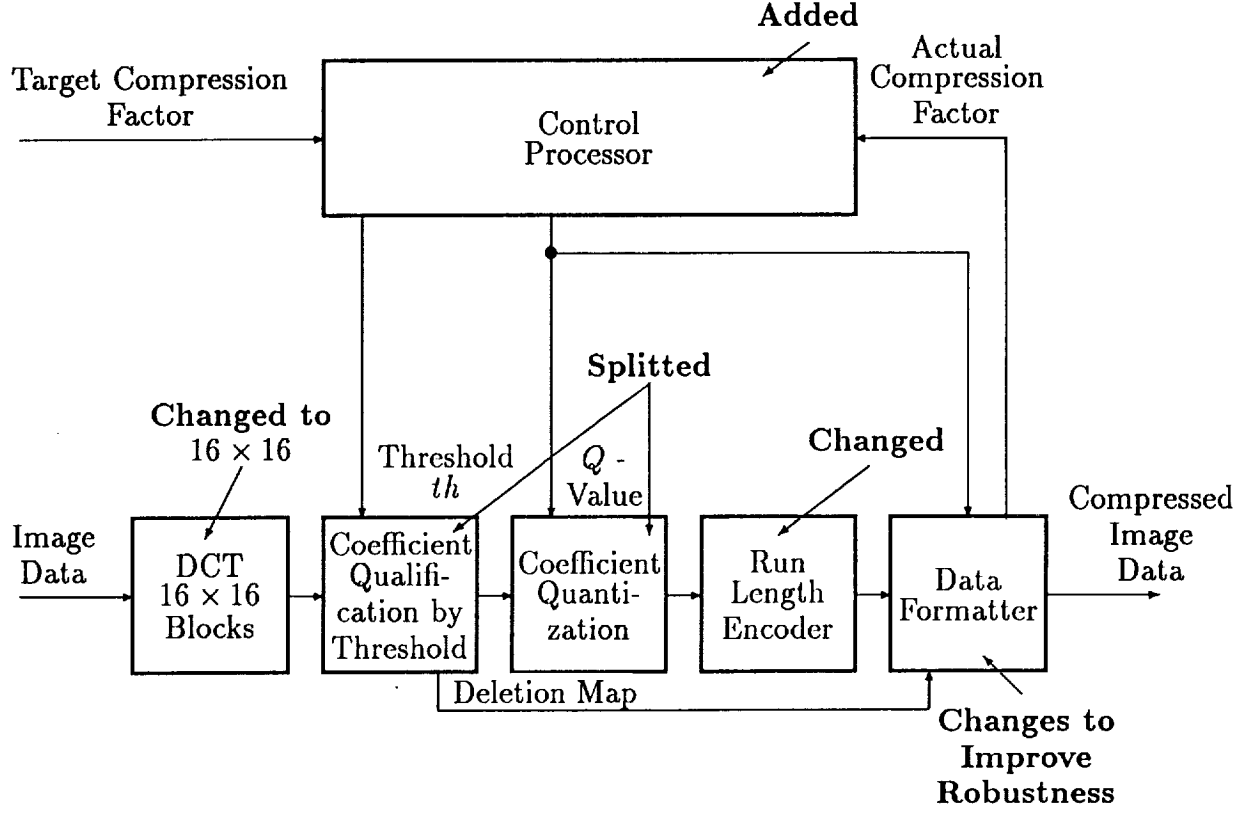


Figure 4: Data/control flow of DISR image data compression scheme

Quantization value Q and threshold th are feedback controlled by the control processor. They are iteratively adjusted until the best approximation of the target compression factor is reached. Iteration time is included in the DISR compression time of less than 0.1 s.

6 Performance

By simulations it has been verified that the IMP S/W implementation delivers JPEG equivalent image quality combined with improved error robustness. Figure 5 shows the signal to noise ratio

$$SNR[dB] = 10 \log \frac{\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_o(n_1, n_2)^2}{\sum_{n_1} \sum_{n_2} (f_o(n_1, n_2) - f_r(n_1, n_2))^2}$$

f_o : pixel intensity of original image
 f_r : pixel intensity of reconstructed image

versus the compression factor c for the well known "Lena" image and a mars surface image which was derived from a viking mission. The DISR H/W implementation shows slightly

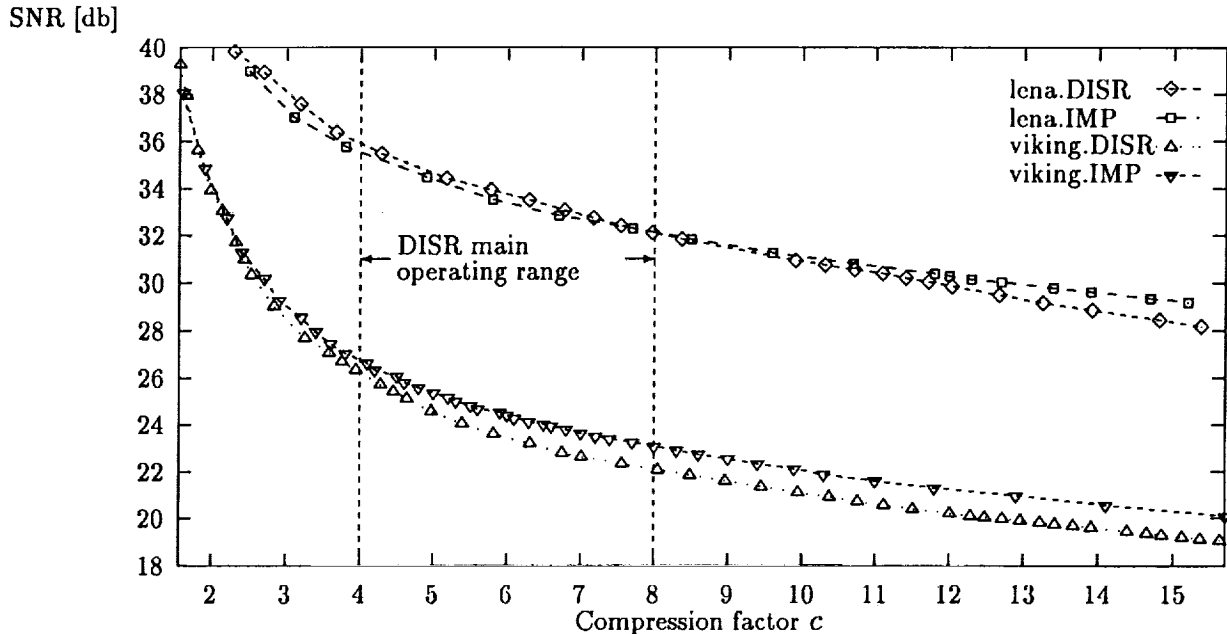


Figure 5: Comparison of IMP and DISR $SNR[dB]$ performance versus c

degraded image quality, but increased error robustness, too. For a compression factor greater than 4 the compression quality expressed by $SNR[dB]$ versus c is degraded to less than 1 dB . But a visual comparison of the decompressed images shows more visible blocking effects. This is caused by suboptimal coefficient quantization and suboptimal redundancy reduction. Still, these slight performance degradations have to be balanced against the substantial higher compression speed.

References

- [1] W.B. Pennebaker, J.L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993
- [2] *The Titan Huygens Probe Descent Imager/Spectral Radiometer*, EID Part B, ESA/ESTEC, 1990
- [3] *Imager for MESUR Pathfinder*, Investigation and Technical Plan, Univ. of Arizona, 1993
- [4] J.R. Parsons, A.G. Tescher, *An Investigation of MSE Contributions in Transform Image Coding Schemes*, SPIE, Vol. 66, 1975
- [5] P. Rüffer, F. Rabe, F. Gliem, *A DCT Image Data Processor for Use on the Huygens Titan Probe*, IEEE IGARSS' 92, Houston, 1992

ADVANCED END-TO-END SIMULATION FOR
ON-BOARD PROCESSING
(AESOP)

Alan S. Mazer

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, California 91109

54-61
458
P. 11

1. Introduction

Developers of data compression algorithms typically use their own software together with commercial packages to implement, evaluate and demonstrate their work. While convenient for an individual developer, this approach makes it difficult to build on or use another's work without intimate knowledge of each component. When several people or groups work on different parts of the same problem, the larger view can be lost. What's needed is a simple piece of software to stand in the gap and link together the efforts of different people, enabling them to build on each other's work, and providing a base for engineers and scientists to evaluate the parts as a cohesive whole and make design decisions.

AESOP (Advanced End-to-end Simulation for On-board Processing) attempts to meet this need by providing a graphical interface to a developer-selected set of algorithms, interfacing with compiled code and standalone programs, as well as procedures written in the IDL and PV-Wave command languages. As a proof of concept, AESOP is outfitted with several data compression algorithms integrating previous work on different processors (AT&T DSP32C, TI TMS320C30, SPARC). The user can specify at run-time the processor on which individual parts of the compression should run. Compressed data is then fed through simulated transmission and uncompression to evaluate the effects of compression parameters, noise and error correction algorithms.

The following sections describe AESOP in detail. Section 2 describes fundamental goals for usability. Section 3 describes the implementation. Sections 4 through 5 describe how to add new functionality to the system and present the existing data compression algorithms. Sections 6 and 7 discuss portability and future work.

2. Design Goals

A few goals are central to the design of AESOP. AESOP must:

1. Be usable enough that scientists and system designers can experiment with their data with little instruction. There must be clear visual feedback as applications execute. The user must be able to easily display algorithm data using a variety of display types.
2. Be easy to augment. It should be easy to integrate executables for which source is unavailable, as well as code written in compiled languages such as C and FORTRAN. Non-programmers should be able to use a high-level interpreted language to add capabilities.

3. Rely on outside development when such is commonly and cheaply available. It should provide for the integration of commercial packages as much as possible.
4. Isolate itself from applications; changes to AESOP must not require that applications be rebuilt or otherwise modified.
5. Provide complete error handling. AESOP must be prepared to handle internal errors, user errors and errors in applications, in a useful way, preserving the current state and providing the user options as much as possible.
6. Coexist well with other executing software. It should be efficient and flexible in use of screen space and other system resources.
7. Be user-customizable in look. The user should be able to choose cosmetic features such as user interface colors, as well as operational defaults, such as which types of displays are automatically enabled.

3. Implementation

The AESOP implementation assumes two simple concepts: **modules**, compiled or interpretable code which performs specific computations, and **algorithms**, module sequences used to implement complete applications. The following sections describe these two concepts in more detail, and then show how they provide a basis for the complete system.

3.1. Modules and Algorithms

Each AESOP module, compiled or interpreted, has a usage type and some number of input and output arguments. Input modules are used to read in files from disk or bring other data into the system which the user can't practically enter from the keyboard. Compute modules perform computational tasks. Output modules are selected at run-time by the user and perform data display. Arguments also have usage types. An input argument is one read by the module; an output argument is a value or data item that the module generates. Update arguments are both read and modified by the module. Each argument also has a data type, as summarized in Table 1.

Table 1 – AESOP data types		
char	char_1d	char_2d
short	short_1d	short_2d
int	int_1d	int_2d
float	float_1d	float_2d
double	double_1d	double_2d
string	string_1d	string_2d
kwd	kwd_1d	kwd_2d

An AESOP algorithm is a sequence of compute modules where the inputs for each module are taken either from the user or from the output of a previous compute module. Algorithms are typically a mixture of compiled and interpreted modules.

3.2. The Dictionary Interface

Figure 1 shows an overview of AESOP implementation. Sections 3.2 through through 3.4 will discuss the major components, beginning with the dictionary interface and continuing with code execution and the GUI.

Dictionaries are ASCII files listing available **modules** (compiled routines, binary executables, interpretable procedures) and **algorithms** (module sequences designed to perform common tasks). AESOP looks for one standard dictionary, "stdlib.dict", to contain generally useful routines for output display, local file formats, etc. Users may define any number of other dictionaries to describe modules and algorithms in specific application areas. AESOP looks for dictionaries in the local directory, with the AESOP executable, and in other directories specified by the user using the AESOP_APPL_DIRS environment variable. Dictionaries can be reread without leaving AESOP to gain access to newly-defined or modified algorithms and modules. Dictionaries can also contain graphics directives specifying how an algorithm is displayed on the screen, including labels and boxes. Dictionary entries have several formats depending on whether they are defining a compiled module, an interpreted PV-Wave module or an algorithm.

Entries for compiled modules have the form:

```
module_type name:label:pathname
```

PV-Wave modules are defined similarly, but with the module inputs and default values following the pathname. Entries for interpreted PV-Wave modules have the form:

```
module_type name:label:pathname:
    arg_use_type1 arg_data_type1 arg_label1[=default],
    arg_use_type2 arg_data_type2 arg_label2[=default], ...
    arg_use_typen arg_data_typen arg_labeln[=default]
```

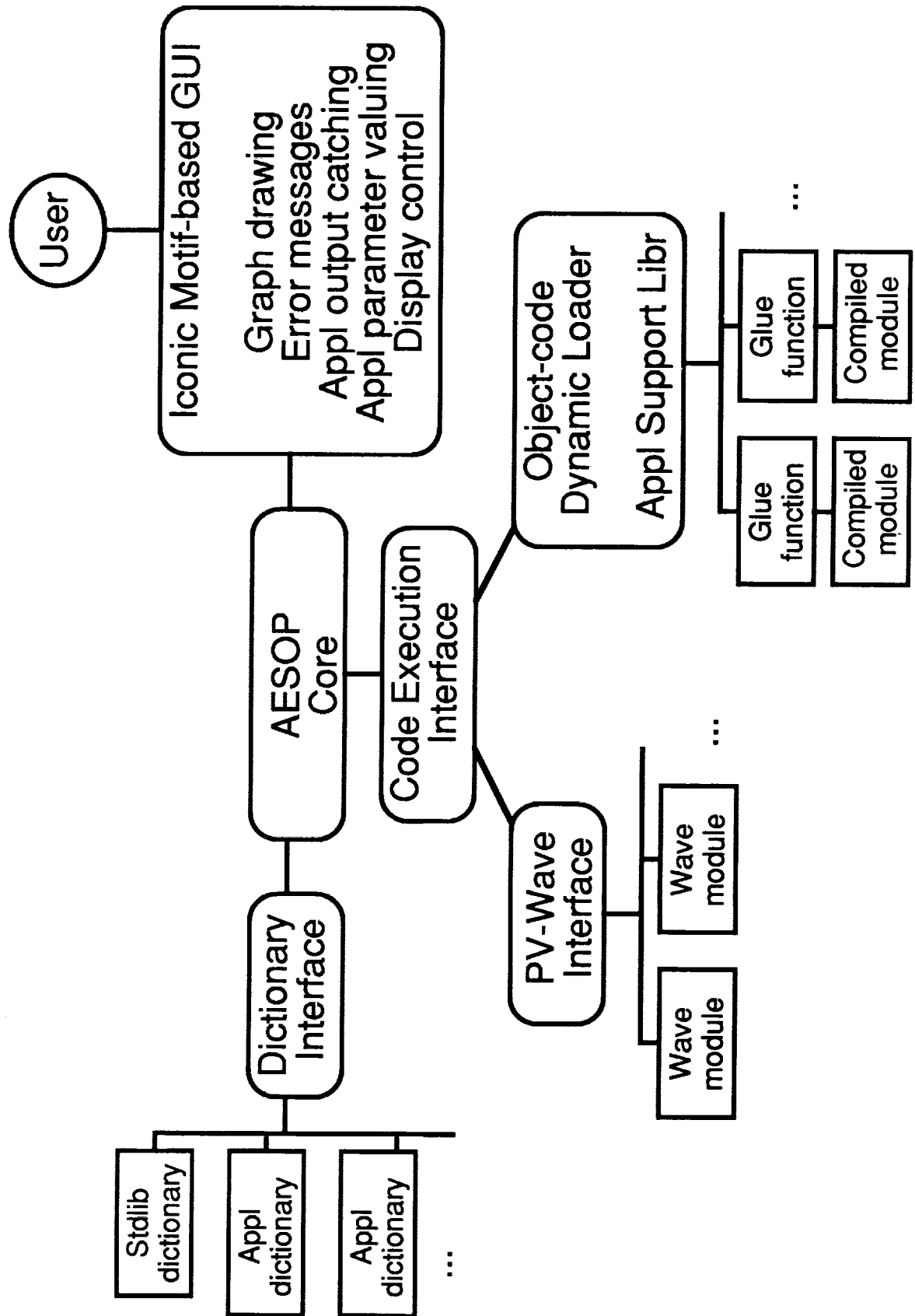
The first line of the entry is similar to the entry for the compiled-module. Subsequent lines list parameters, separated by commas, where each parameter has a use type, data type and prompt. Initial values may be specified by following the prompt with an equal sign (=) and the value. Scalars are considered user options automatically; higher-dimensioned parameters are retrieved from previously-executing modules. Type conversions are implicit.

Dictionary entries for algorithms have the basic form:

```
algorithm name:label:module1 module2 ... modulen
```

Extensions to this basic syntax allow the user to group modules in labeled boxes and to lay these boxes in any direction.

Implementation Overview



3.3. The Code Execution Interface

AESOP provides access to two different types of modules: interpreted modules written in the PV-Wave command language and compiled modules written in C or another high-level language. Both types of modules have "glue functions" which are called by AESOP and call the module code in turn. This approach isolates the details of executing application code from AESOP internals.

In the case of compiled code, glue functions are programmer-written and allow AESOP to call executables for which source code is unavailable, as well as routines written in languages other than C. The glue function, written in C, creates local storage for use by the function and defines parameters in a manner AESOP can understand. AESOP calls these glue functions using dynamic loading, further isolating application routines from AESOP itself. The parameter definition interface is simple, using keywords and program-callable functions for optional capabilities, allowing the interface to be extended in the future without requiring modification of currently-integrated code. Glue functions for compiled modules take a single argument, an initialization flag. When an algorithm is selected, AESOP calls the glue function for each compiled module in the algorithm with the initialization flag set to 1. At this time each module uses the AESOP `def()` function to describe its parameters where `def()` is defined:

```
def(char *prompt, enum use_type use, enum data_type type,  
    void *local_addr, char *kwds[], int num_kwds, int option1,  
    int option2, ..., 0)
```

The glue function will be called a second time, with the initialization flag 0, when the module is actually executed. The `kwd` data types provide a simple way to restrict the user's choice of values. Glue functions can indicate an error in either their initialization or execution parts by returning -1, causing AESOP to stop algorithm execution with that module.

For PV-Wave modules, a generic glue function is supplied by AESOP. Since PV-Wave modules have their parameters defined in the dictionary, their glue function need only be called at execution time, when it creates temporary files needed to communicate with PV-Wave, instructs PV-Wave to read necessary data, and invokes the PV-Wave procedure. Module parameters listed in the dictionary and valued by the user before the run are passed in as arguments to the procedure. The AESOP-Wave interface uses temporary files and PV-Wave's `cwavec()` facility. The AESOP-Wave interaction is transparent to the developer and user.

When an algorithm is loaded, AESOP automatically matches up non-user-specifiable parameters. It does this by comparing the names of module outputs with the names of inputs from subsequent modules and assigning to each possible matchup a score. This scheme will probably need to be refined in the future. At the moment, close attention must be given to an algorithm in development to make sure AESOP is attaching inputs to outputs as expected. AESOP uses dimensionality and data type to reduce the potential for error. Nevertheless, simple generic names are best, for example, "output image" rather than "decompression output". In the latter case, a subsequent module expecting "input image" might get connected up with some other "image" in the system, rather than the more ambiguous "decompression output". Once all the connections have been made, AESOP uses the PV-Wave or dynamic load-interface as necessary to execute each module in turn. AESOP ensures before each

module is executed that the inputs to the module are available, either because the user explicitly specified them or because they were generated by a previous module in the algorithm. Signal handlers are installed to catch memory usage errors in applications. If AESOP detects such an error it stops execution of the module, restoring itself to its state before execution started.

3.4. The GUI

The usability goals described in Section 2 are met in part by a graphical interface. Most user interactions can be done with the mouse. The current status of the system is graphically displayed. Options prohibited in a specific context are hidden until needed to avoid confusion. The implementation is divided into 5 general parts: graph drawing, error messages, application output catching, application parameter valuing and display control.

The graph drawing section presents algorithms selected as dataflow diagrams. Graph drawing is done using X11/Motif, with application modules represented by boxes and connected with arrows in a single-stream pipeline. Modules may be grouped and groups labeled. Groups may be oriented in any direction, clearly distinguishing different parts of an algorithm. Grouping, labeling and orientation are optional and taken from the algorithm specification in the dictionary. When algorithms execute, module boxes are highlighted to show progress. Since for large algorithms the graph area may not be large enough to show all the modules, the graph area scrolls itself to keep the currently executing module visible.

The error messages section alerts the user to AESOP-discovered error conditions using popup windows. AESOP detects 39 different error conditions, including fatal memory usage errors in application modules. AESOP shows a popup window describing the condition and then waits for user acknowledgement before continuing. Error messages printed by an application module are also displayed in popup windows.

Non-error output from an application module is caught and optionally displayed in its own window. When a module tries to send informational messages to the user, AESOP grabs that output and, if the user has requested diagnostic output, displays it in a window created for that purpose. Otherwise the output is discarded. AESOP can maintain a separate window for each module, and switch between them as the different modules execute. This capability allows the user to choose which parts and how much of the execution details to view, and simplifies debugging during module development.

The application parameter valuing section allows the user to give values to optional and required module parameters using popup windows. Both interpreted and compiled modules may take parameters. The user specifies a value for a module parameter using the pulldown menu attached to the module in the graph. AESOP lets the user enter scalar numerical quantities or choose items from lists using the keyboard. For larger parameters like input images the user selects a module to use to read in the required data. Such modules are typically defined in the standard library but are otherwise similar to application modules.

Finally, AESOP allows the user to monitor module inputs and outputs using a variety of display types. When AESOP starts it builds a list of all output modules listed in the dictionaries. It then sorts the modules based on data type and the dimension of the primary input(s), where a primary input is defined as an input such that no other input has a larger number of dimensions. When the user requests display of a module input or output using a

module's menu, AESOP allows the user to select a parameter to display and then presents a list of output modules suitable for displaying that particular type of value. Alternatively the user can add a display using the **Displays** menu. AESOP allows the user to specify the dimensionality of the data and the type of display to create using the menu, and then presents a list of module parameters displayable with that type of output module. Since some display modules will take inputs other than the data to display, AESOP prompts the user for needed information; in the case of non-scalar inputs, it offers choices from among the data items currently available in the system. These capabilities are provided automatically by AESOP and do not depend on the algorithm writer. The **Displays** menu also allows users to change or remove displays. PV-Wave has been used to implement most of the current output modules.

Figures 2 and 3 show AESOP adding noise to a JPEG-compressed image and the resulting output with no error correction.

4. Programming Environment

Adding functions or subroutines written in C, FORTRAN and other compiled languages requires only writing the glue function and adding the name and object file pathname to a dictionary. Glue functions for compiled modules have two parts: the initialization part which defines parameters using AESOP's `def()` function, and an execution part to call the compiled function. Glue functions should return `-1` on discovering a fatal error and `0` otherwise. Error messages should be written to `stderr` and informational messages to `stdout`. The dictionary entry for the DCT compute module declares the type of the module, its name, the label to use on the graph, and the pathname of the glue-function object:

```
compute_module jpeg_dct:DCT:lib/rpc.so
```

The glue function must be compiled and linked with the functions it calls into an executable, with a ".so" extension. For SunOS one would use:

```
acc -c -pic glue_funcs.c
ld -o library.so glue_funcs.o funcs_to_add.o
```

Generally useful functions should go into the standard library ("stdlib.dict"). Other functions can be listed in application dictionaries. Once the module has been specified in one or the other type of dictionary it's available for use.

Adding code from PV-Wave and other command-line-based packages is similar to adding compiled code, except that parameters are declared in the dictionary rather than using a glue function:

```
output_module flick2:Alternate Two Images:flick2.pro:
    input u_char_2d First Image, input u_char_2d Second Image,
    input int Iterations=20, input float Wait=0.3
```

Algorithms are added by simply defining them in the dictionary as an ordered list of module names:

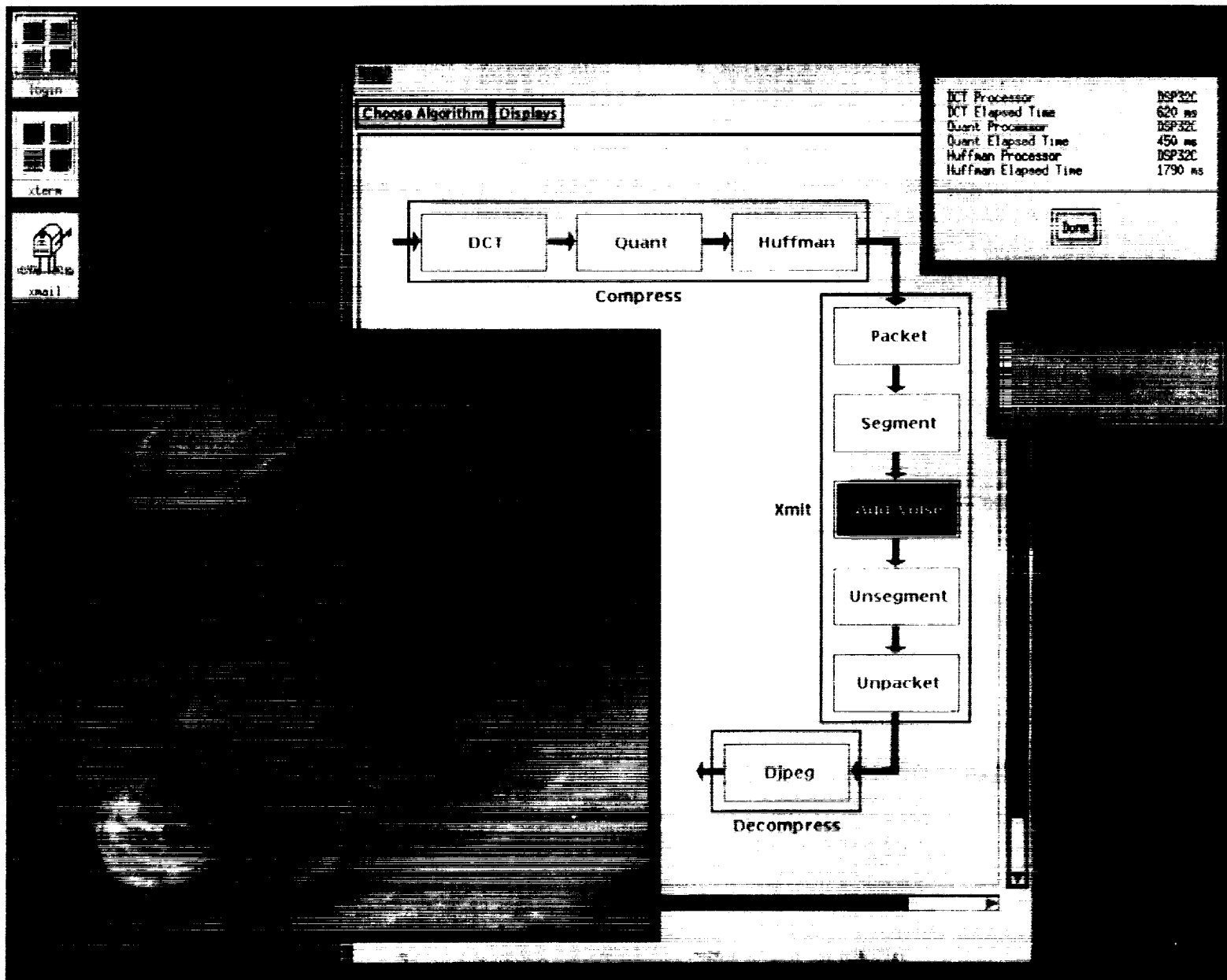


Figure 1 - AESOP execution of JPEG algorithm during downlink simulation

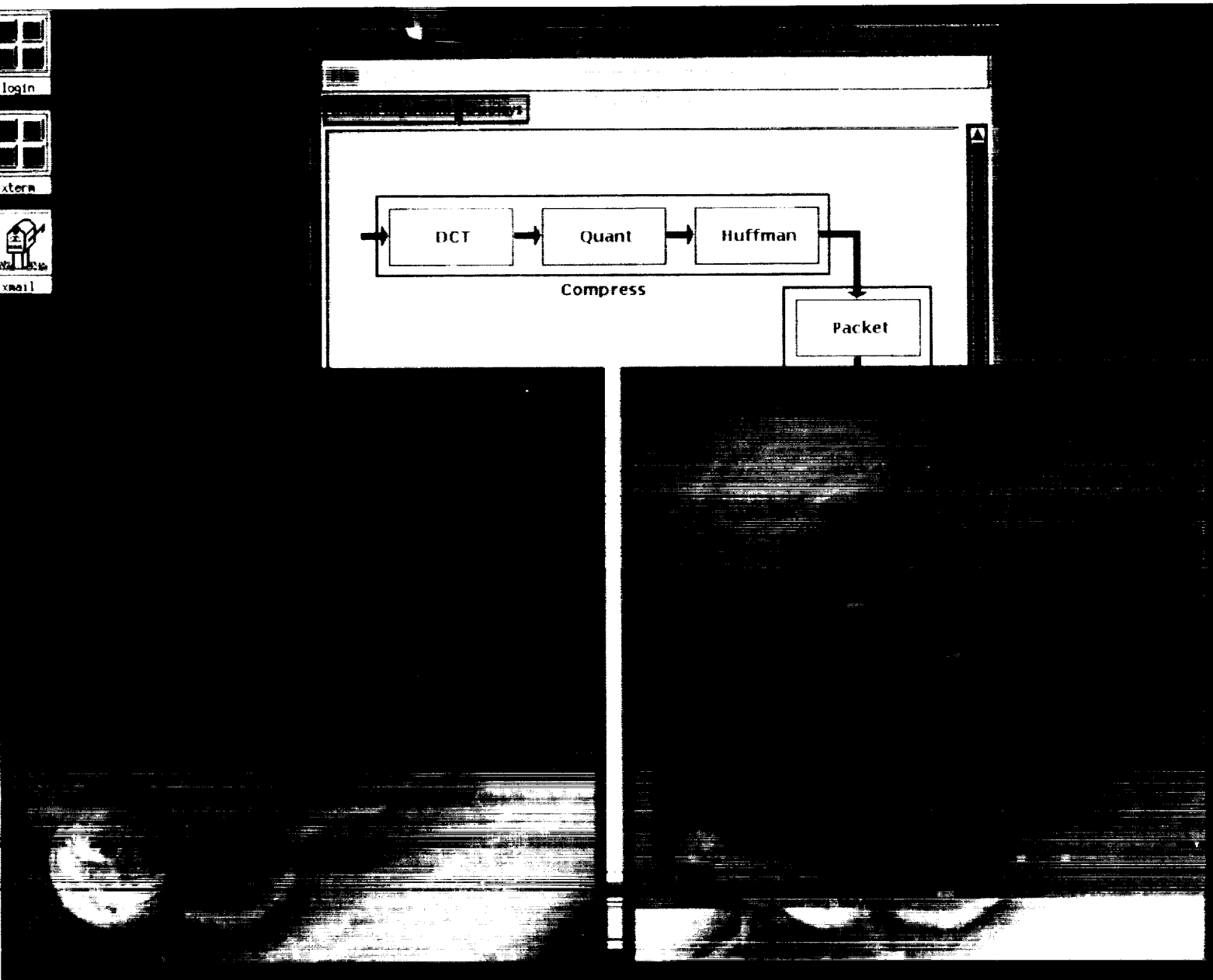


Figure 2 — Image as hypothetically sent and received with random single-bit errors (30,000 bit interval, no channel coding)

```
algorithm jpeg:JPEG:jpeg_dct jpeg_quant jpeg_huff jpeg_decomp
```

The dictionary syntax allows the user to group modules in labeled boxes and to lay these boxes in any direction. A group is introduced using a vertical bar (|) followed immediately by the label for the group, a direction indicator (>, <, ^ or !), a list of space-separated modules forming the group, and the direction indicator again. The algorithm shown in Figure 2 was defined using:

```
algorithm jpegendtoend:JPEG End-to-end:
  |Compress>jpeg_dct jpeg_quant jpeg_huff>
  |Xmit!packet segment addnoise unsegment unpacket!
  |Decompress<jpeg_decomp<
```

5. Data Compression Applications

Application development for AESOP so far has centered on data compression, but includes simulation of flight-to-ground downlinks. Thus there are application modules not only for various types of compression (JPEG, Rice, one- and two-dimensional wavelet compression) but also for packetization, segmentation, channel coding and noise simulation, providing a true end-to-end view from in-flight data acquisition to the reception of transmitted data on the ground. Supporting the end-to-end simulation of compressed data transmission are a number of computational capabilities (packetization, segmentation and channel coding, and noise simulation) as well as output types.

The packetization routine takes compression output and a set of packet lengths in bits, and breaks the output into packets at the specified bit boundaries. Currently, variable length packets are formed such that each packet holds 8 lines of compressed image data. This approach simplifies recovery should an entire packet be lost since the location of a packet in an output image can be coded in the header, and the break is guaranteed not to occur in the middle of a pixel. An inverse procedure takes incoming packets and recombines them into a single bit stream for decompression.

Because channel coding requires fixed-length chunks of input data, packets are themselves grouped into interleaved segments of uniform length; segments are packed into frames. The interleave factor is an option with a default value of 8. Segmentation currently uses Reed-Solomon coding for optional error correction. The inverse procedure unencodes the data and restores the original input packets. Some diagnostic information (error counts, frame statistics) is available using **Show diagnostics** on the module's menu.

A noise simulation module takes compressed, packetized, segmented data and flips bits on a random interval. The user can specify the mean number of bits between errors, or turn off noise simulation altogether. Better noise models are being developed.

In addition to many output modules in the standard library for reading, writing and displaying various data types, of special interest for data compression algorithms are "Showboth", which allows a user to see two different images side by side, "Flick2", which alternates two images rapidly in the same window using a user-chosen interval and number of iterations, and "Imagediff", which displays the difference of two images using a user-chosen multiplication factor. These are currently restricted to byte input images. Other modules compute signal-

to-noise ratios for most vector and image data types.

6. Portability

AESOP currently runs on Sun SPARCstations using SunOS 4.1.3 and Motif. While PV-Wave is not required, support for it is built in and the current dictionaries use it for image display. Operating system dependencies are minimal. AESOP is written in ANSI C. AESOP uses dynamic loading to execute compiled modules, which is available on AIX 3.2, HP-UX 8.0 and VMS 5.0 in addition to SunOS.

7. Future Work

The foundation is in place, but work remains to be done. AESOP currently relies heavily on PV-Wave for output display; other packages need to be integrated for portability. More output types, particularly for one-dimensional data, need to be implemented. Support for application-defined data structures would be useful. Some applications may have trouble with AESOP's redefinition of the C `write()` routine. Determination of graph connectivity will eventually need enhancement. More control over output displays needs to be added.

8. Acknowledgements

The research described in this paper was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the National Aeronautics and Space Administration, Office of Advanced Concepts and Technology.

PERFORMANCE CONSIDERATIONS FOR THE APPLICATION OF THE LOSSLESS BROWSE AND RESIDUAL MODEL

Walter D. Abbott III
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943
(408) 656-2101
(408) 656-2760 (fax)
abbott@ece.nps.navy.mil

55-61
459
P. 12

Robert T. Kay
Department of Electrical and Computer
Engineering
Naval Postgraduate School
Monterey, CA 93943

Ron J. Pieper
Department of Electrical and Computer
Engineering
Naval Postgraduate School
Monterey, CA 93943

Abstract. A hybrid lossless compression model employing both the (lossy) JPEG DCT algorithm and one of a selection of lossless image compression methods has been tested. The hybrid model decomposes the original image into a low-loss quick-look browse and a residual image. The lossless compression methods tested in the model are Huffman, arithmetic, LZW, lossless JPEG, and diagonal coding. For both the direct and the hybrid application of these lossless methods, the compression ratios (CRs) are calculated and compared on three test images. For each lossless method tested the hybrid model had no more than a nominal loss in compression efficiency relative to the direct approach. In many cases, the hybrid model provided a significant compression gain. When used in the hybrid model, lossless JPEG outperformed the other lossless methods over a broad range of browse image qualities.

1. Background

In many practical situations involving images, a small degree of error in the pixel values can be tolerated without a significant effect on the display. This suggests that there are advantages to a decomposition of images into a lossy component, or browse component, and an error or residual component. The decomposition of the original image into browse and residual images gives an end-user the ability to browse an image and determine whether the residual image should be transmitted and added to the browse image to reproduce the original image. This feature is not available with any direct lossless compression method. A hybrid compression model employing the (lossy) JPEG DCT algorithm with the lossless diagonal coding scheme has recently appeared in the literature [1].

Some of the standard lossless compression methods are Huffman, arithmetic, the Ziv and Lempel algorithms, predictive encoding, bit-plane encoding, and run-length encoding [2]. Each of these compression methods have many variations which are reported in the literature. Another lossless compression method is lossless JPEG which utilizes a combination of predictive encoding and Huffman [3]. A non-standard lossless compression method is diagonal coding [1]. Diagonal

coding is a type of lossless variable length encoding designed to take advantage of the Laplacian distribution characteristic of the residual image. For efficient compacting of the coded bit stream, a special C source code program was written that operates at the bit level [4]. Operating at the byte level would destroy any advantages of this coding method. Lossy compression methods consist primarily of the Joint Photographic Experts Group (JPEG) algorithm [5] and fractal encoding [6].

2. The Lossless Hybrid Model

The hybrid model utilizes both a lossy and a lossless image compression technique to produce an overall lossless image compression. Such an arrangement takes advantage of the high compression ratios achieved by the lossy methods and the error-free compression of the lossless methods. The image is first compressed using a lossy compression method. The lossy compressed image is decompressed and compared on a pixel-by-pixel basis with the original image. The decompressed image is termed the browse image as it can be used to browse an image for suitability for the application intended. The difference between the original image and the decompressed image is termed the residual image. The residual image is compressed using a lossless compression method. The compressed browse and compressed residual images can be appended for calculating overall compression. The forward process described here and the corresponding reverse process are presented in Figures 1a and 1b.

Because of the general acceptance and effectiveness of (lossy) JPEG [3], all the results from our hybrid model investigations presented here use this method to produce the browse images. A similar investigation used fractal compression with LZW compression [7].

Our test results indicated that it is not feasible, in terms of compression overhead, to use secondary compression to significantly compress either the compressed browse or compressed residual. In most cases tested, secondary compression resulted in expansion of the compressed image file size [4]. As a result, secondary compression was not included in the hybrid lossless compression model presented here.

One compression measure used to gauge performance is the compression ratio (CR) defined as [8, p. 10]:

$$CR = (1 - (\text{Compressed Image Size} / \text{Original Image Size})) \times 100. \quad (1)$$

The overall compression ratio achieved by the hybrid lossless compression model is a combination of the compressed browse image CR and the compressed residual image CR. Application of Equation (1) to browse, residual, and overall compression ratios leads to:

$$CR_{\text{overall}} = [CR_{\text{browse}} - 50] + [CR_{\text{residual}} - 50] \quad (2)$$

where CR_{browse} and CR_{residual} are the compression ratios of the compressed browse and residual images.

3. The Test Images

The hybrid model (Figures 1a, 1b) was tested and evaluated using three 8-bit, 256x256 pixel images in raw pixel grey map format. The three images (Figure 2) were selected based on their structurally different pixel distributions or histograms (Figure 3).

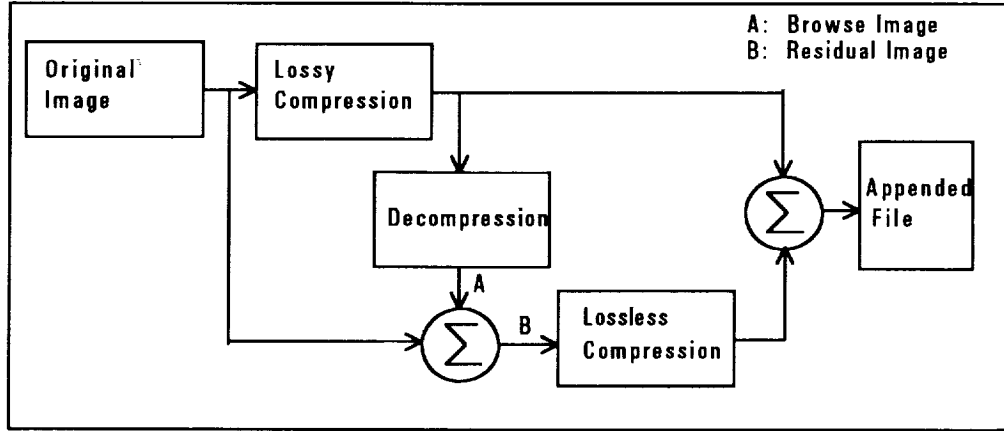


Figure 1a: Lossless Hybrid Model Compression.

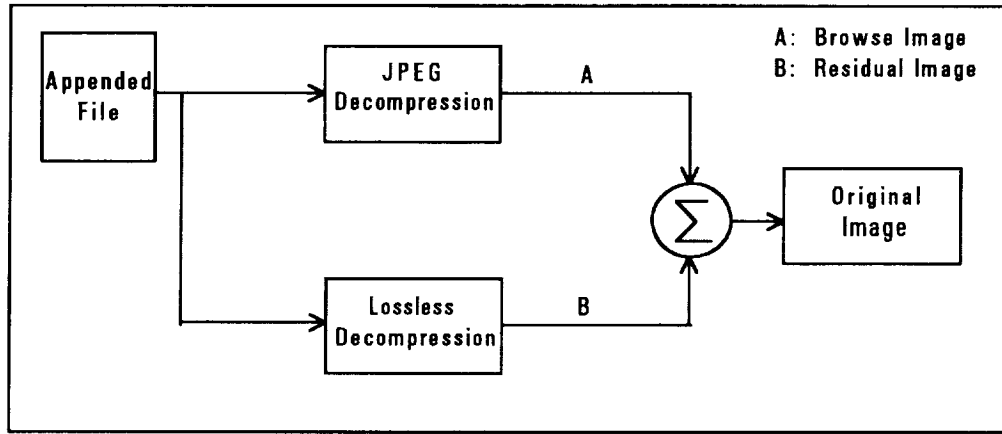


Figure 1b: Lossless Hybrid Model Decompression.

The lossy JPEG algorithm used in the model was developed by Andy C. Hung at the Portable Video Research Group (PVRG), Stanford University [5]. The quality factor used when compressing an image determines the amount of compression achieved and the resolution of the image when it is decompressed. The higher the quality factor, the greater the compression and the less the resolution upon decompression. Figure 4a graphically displays the quality factor versus compression ratio achieved for the three test images. One common measure of the resolution of the decompressed image as compared to the original image is termed the root mean square error (e_{rms}) as defined by:

$$e_{rms} = \frac{1}{N} \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |g(x,y) - f(x,y)|^2 \right]^{0.5} \quad (3)$$

where, for $N \times N$ pixel images, $f(x,y)$ is the array of pixel values for the original image while $g(x,y)$ is the array of pixel values for the decompressed image [9, pp. 256-257]. Figure 4b graphically displays a plot of quality factor versus e_{rms} for each of the three test images. As the quality factor is increased, the e_{rms} of the decompressed image decreases as expected. The decompressed test image LENA is displayed in Figure 5 after compression at various quality factors. Note that as

the quality factor increases, the resolution of the decompressed image decreases. At quality factors greater than 250, the decompressed image begins to exhibit distinct blockiness due to the processing of 8x8 pixel blocks by the JPEG algorithm.

The residual image resulting from the pixel by pixel differences in the original image and the decompressed image exhibits a Laplacian distribution with a mean of zero [2, p. 60]. The residual image distribution, or histogram, has a reduced variance compared to the original image and is also significantly less correlated. The shape of the residual image histogram is dependent upon the quality factor used to compress the original image using lossy JPEG. As previously discussed, the higher the quality factor used, the more compression achieved; however, the decompressed image will less resemble the original image. This results in a residual image containing a wider range of pixel values. As a result, the residual image histogram will exhibit a wider Laplacian distribution. Figure 6 displays residual image histograms of LENA for various quality factors. Note that as the quality factor used to compress the original image of LENA is increased, the distribution of the corresponding residual image widens.

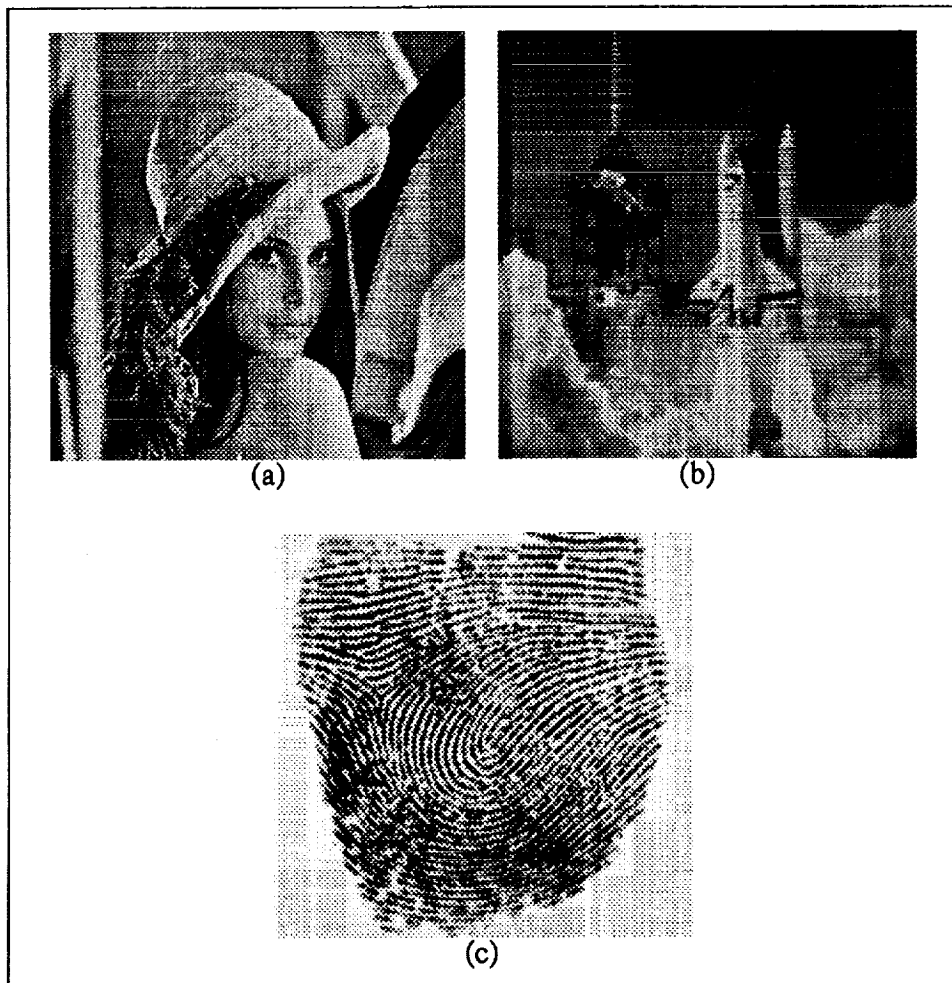


Figure 2: Three Test Images (a) LENA, (b) SHUTTLE, (c) FINGERPRINT.

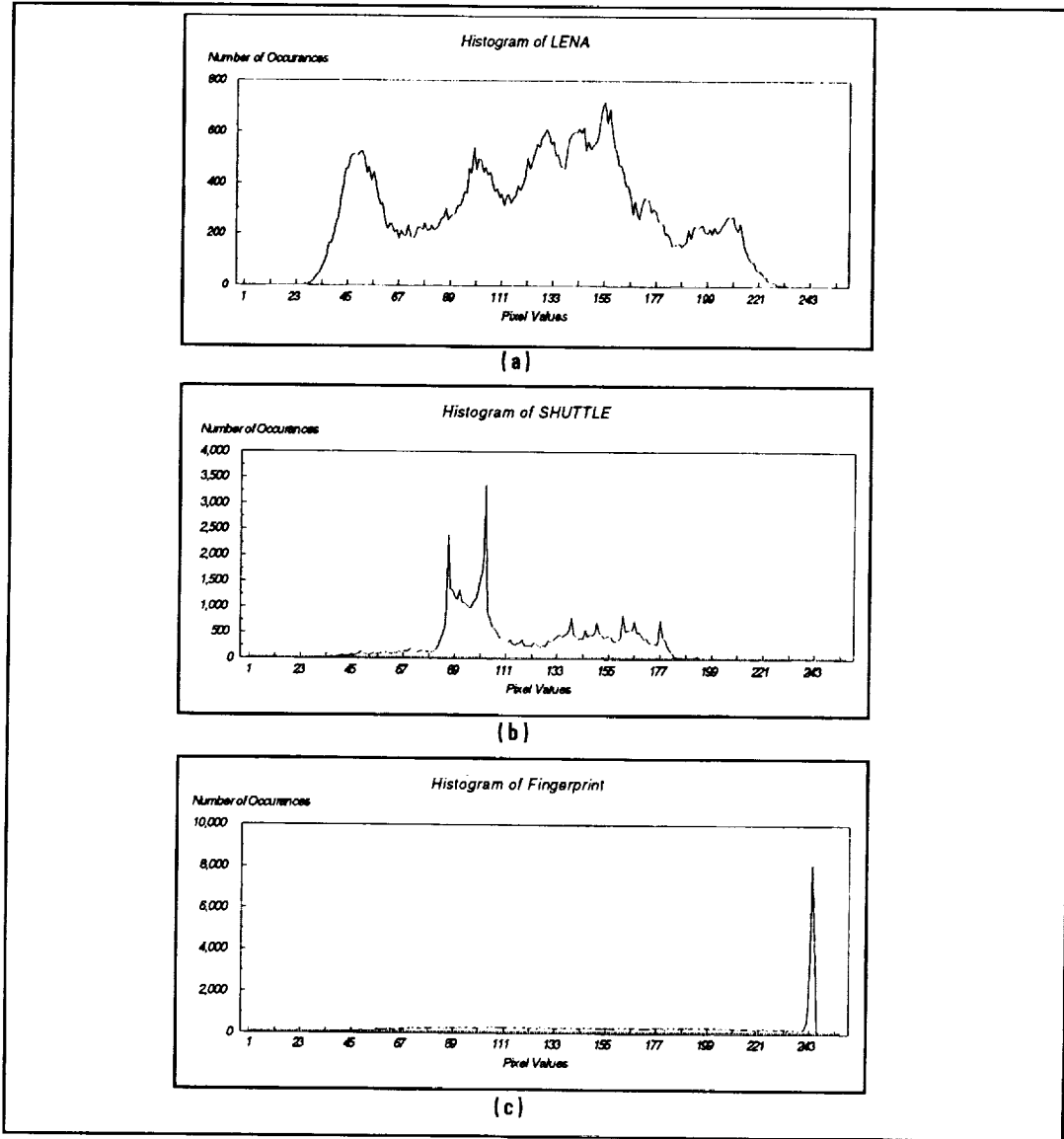


Figure 3: Histograms of the Three Test Images (a) LENA, (b) SHUTTLE, (c) FINGERPRINT.

4. Testing the Lossless Hybrid Model

The hybrid model (Figure 1a) was tested using lossless compression techniques previously mentioned. Huffman, arithmetic, diagonal, and lossless JPEG were used to compress the residual image ((B) shown in Figure 1a). A comparison between the compression results achieved by the direct lossless compression methods and the hybrid model is graphically displayed in Figures 7a, 7b, and 7c for each of the three test images at various quality factors. The corresponding results for LZW are summarized in Figure 8. For ease of reading, it should be noted that the right-most 3-D bar in each column represents the compression achieved with that particular direct lossless compression method (not using the hybrid model). The graphical results of using diagonal coding

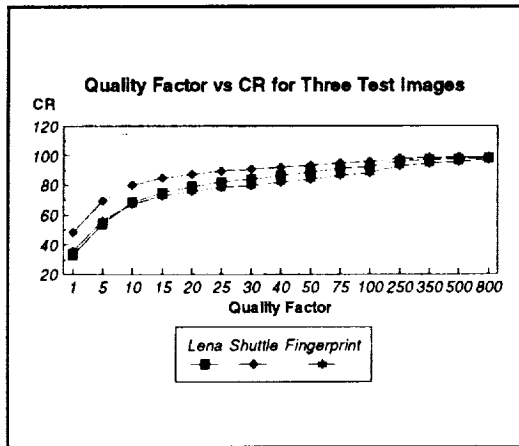


Figure 4a: Comparison of Quality Factor vs CR for the Three Test Images.

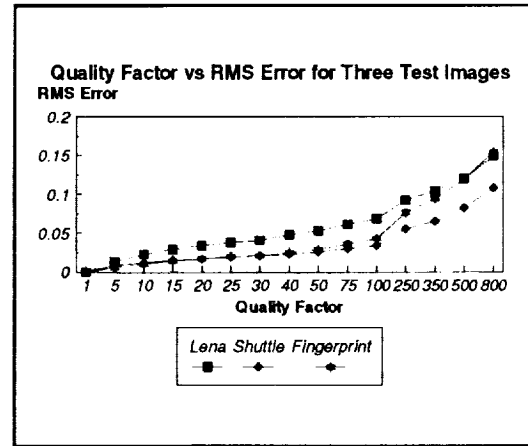


Figure 4b: Comparison of Quality Factor vs e_{rms} for the Three Test Images.

in direct lossless compression is limited to a CR of -30% for each of the images due to the degree of expansion diagonal coding produces when used in the direct compression application. Diagonal coding produced CRs of -76%, -111%, and -144% when used to compress LENA, SHUTTLE, and FINGERPRINT directly. In all cases, the hybrid model achieved greater compression ratios on all three test images than did the direct lossless compression methods with the exception of the direct application of the lossless JPEG method. From a comparison of Figures 7a, 7b, 7c, and Figure 8, LZW does not appear to be a wise choice for lossless compression in the hybrid model. LZW does not surpass the performance of the other methods for any quality factor tested. The residual images do not contain long repetitive strings of pixel values which are necessary for LZW to achieve high compression results. This is not surprising since the LZW method is designed primarily for compressing text, not visual graphics [8, pp. 23-24]. For this reason the LZW results will not be included in the discussion of comparisons which follow.

The CR for diagonal coding is not superior to the set of lossless methods at any quality factor (see Figures 7a, 7b, 7c); however, it does achieve close to the same compression results as Huffman, arithmetic, and lossless JPEG at some quality factors. As the quality factor used to compress the original image is increased, the compression achieved using diagonal coding decreases. This is due to the residual image distribution widening, thereby resulting in longer diagonal codes. At some point, diagonal coding will result in the expansion of the residual image file size. Diagonal coding resulted in an expansion of the residual image size when used to compress FINGERPRINT at a quality factor of 500 (see Figure 7c). It may be noteworthy that the execution time for the diagonal coding method was qualitatively observed to be shorter relative to the execution times for the computationally intensive Huffman, arithmetic, and lossless JPEG algorithms.

Using only the CR as the criterion for comparison, the results indicate that for low quality factors (≤ 50) arithmetic coding is the best choice for lossless compression of the residual images while at higher quality factors (> 50) lossless JPEG is the best choice. Due to the wide diversity in

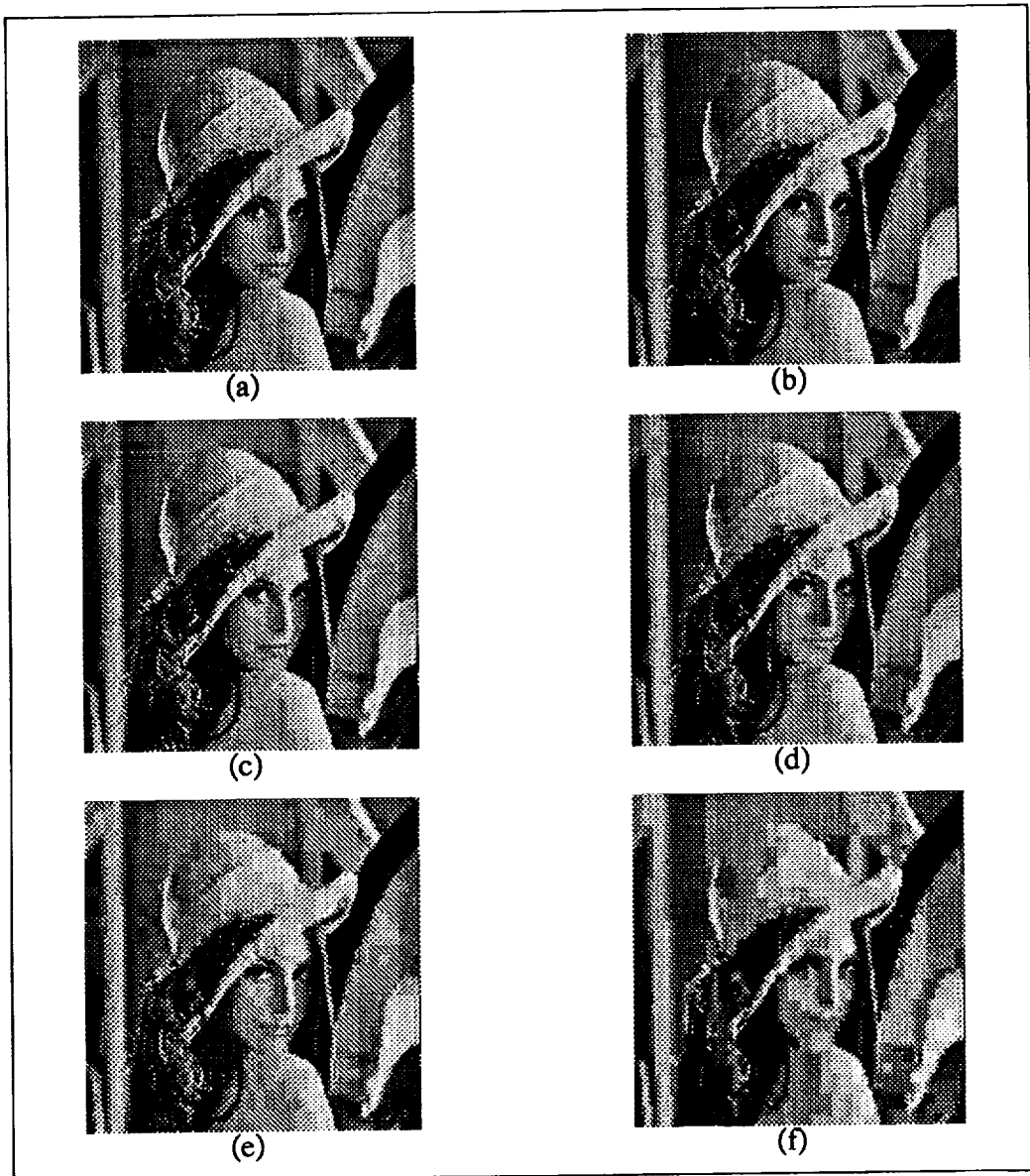


Figure 5: Decompressed LENA at Various Quality Factors (a) Original Image, (b) Q=100, (c) Q=250, (d) Q=350, (e) Q=500, (f) Q=800.

the histograms of the images tested, the observations made here regarding hybrid model performance would ostensibly be qualitatively applicable to a large host of images.

5. Additional Performance Considerations of the Hybrid Model

The hybrid model, using the lossless JPEG, achieved a lower CR on LENA and SHUTTLE than did the direct application of the lossless JPEG; however, the model did achieve a greater CR than direct lossless JPEG on FINGERPRINT at quality factors of 50 and 100 (see Figure 7c). Nonetheless, the hybrid model enjoys the advantage of producing a compressed

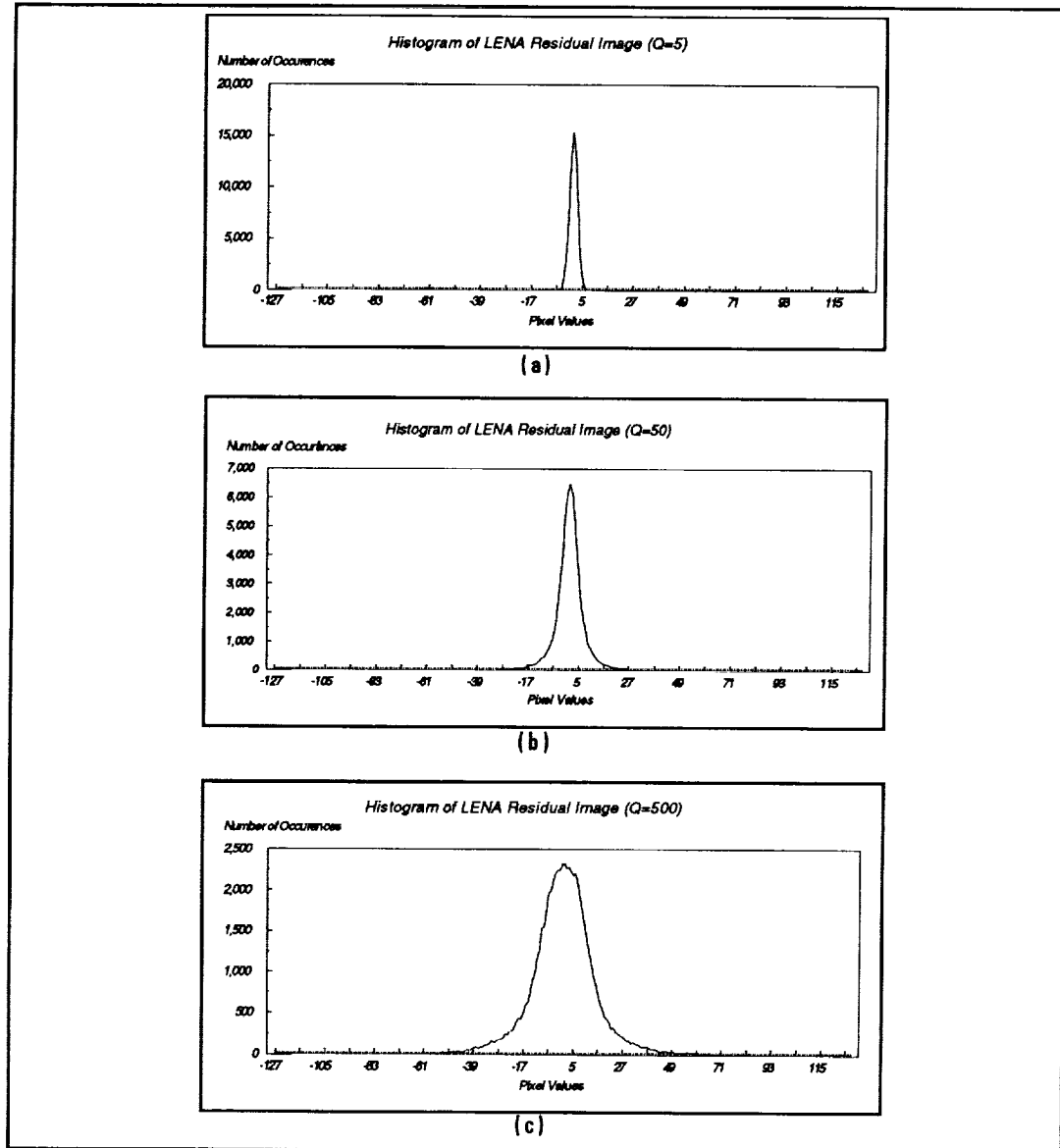


Figure 6: Residual Image Histograms of LENA (a) Q=5, (b) Q=50, (c) Q=500.

browse image which is significantly more compressed than the direct lossless JPEG compressed image. For instance, using a quality factor of 100 to compress LENA produces a quick-look lossy compressed browse image with a file size of 4823 bytes (compression ratio of 92%). The best lossless JPEG predictor algorithm produces a direct lossless compressed file size of 43322 bytes (compression ratio of 34%) (see Figure 9). The Q=100 LENA browse image produces an image that is visually lossless with no visual distortions (see Figure 5). If a lossless image is desired then the residual image of 40353 bytes can be transmitted and added to the browse image to produce an exact replica of the original image.

As previously discussed, the quality factor will impact the Laplacian distribution of the residual image. As seen from Figure 9 for LENA, the compressibility of both the browse and

residual images depend on the quality factor. At low quality factors, minimal compression is achieved on the browse image; however, the residual image becomes highly compressible. As the quality factor is increased, the browse image is more compressible, but the residual image compresses less. These observations also apply to SHUTTLE and FINGERPRINT [4]. Since the overall lossless image is the sum of the compressed browse and residual image data (see Equation 2), achieving maximum overall compression would ostensibly depend on finding some optimal quality factor. In this section, we will examine this issue as well as the sensitivity of the overall CR to the quality factor for the images chosen.

Figures 10a, 10b, and 10c show the overall CR versus quality factors using the hybrid model on LENA, SHUTTLE, and FINGERPRINT respectively. Consistent with the conclusions reached at the end of the previous section, the focus of the comparisons will now be on the application of the arithmetic algorithm and lossless JPEG in the hybrid model. Note that for sufficiently high quality factors the lossless JPEG outperforms arithmetic. Under these conditions, the JPEG predictor is better able to accurately predict pixel values for residual image distributions and therefore produces higher compression ratios. This ostensibly is a result of a higher 2-D correlation of pixel values within the corresponding residual images at higher quality factors (see Figure 4b). As seen from Figures 10a, 10b, and 10c, for quality factors greater than approximately 50, the arithmetic method becomes less effective as the quality factor increases. At the higher quality factors, lossless JPEG achieves asymptotically higher compression ratios. Except at very low quality factors, the test results show that the overall compression ratio achieved by the hybrid model, when using lossless JPEG to compress the residual image, is relatively insensitive to the quality factor used to compress the original image. Therefore the data suggests that for the hybrid JPEG case, the trade-offs which dictate the best JPEG quality factor can be limited to subjective browse image quality and the associated browse compression ratio, but not the overall compression ratio.

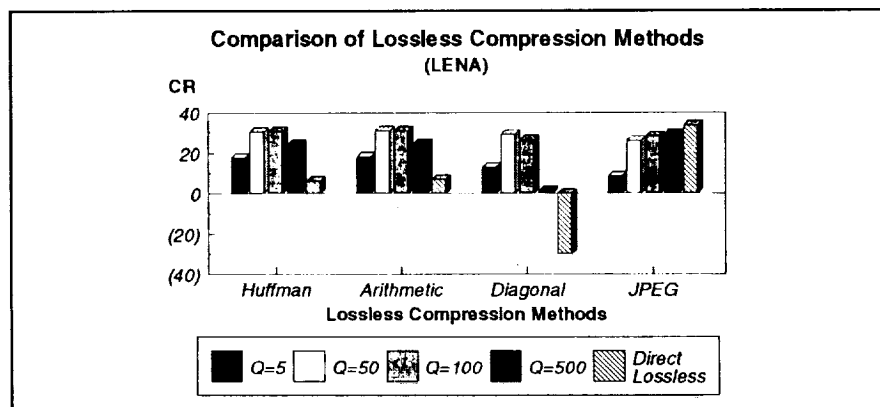


Figure 7a: Comparison of Hybrid Model with Lossless Compression Methods for LENA at Various Quality Factors.

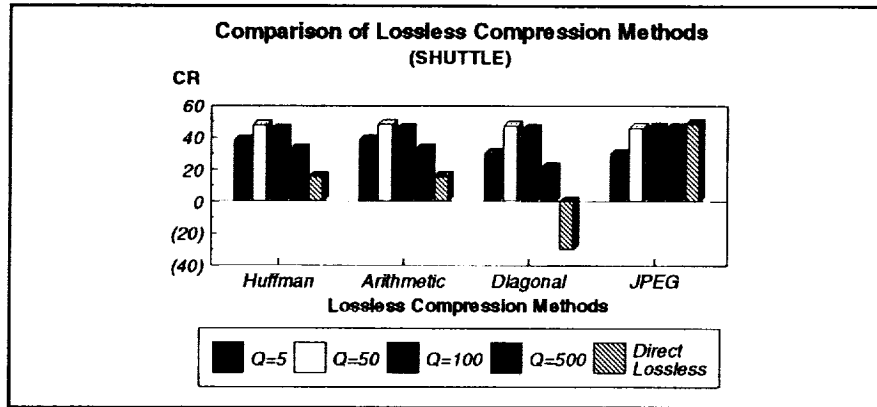


Figure 7b: Comparison of Hybrid Model with Lossless Compression Methods for SHUTTLE at Various Quality Factors.

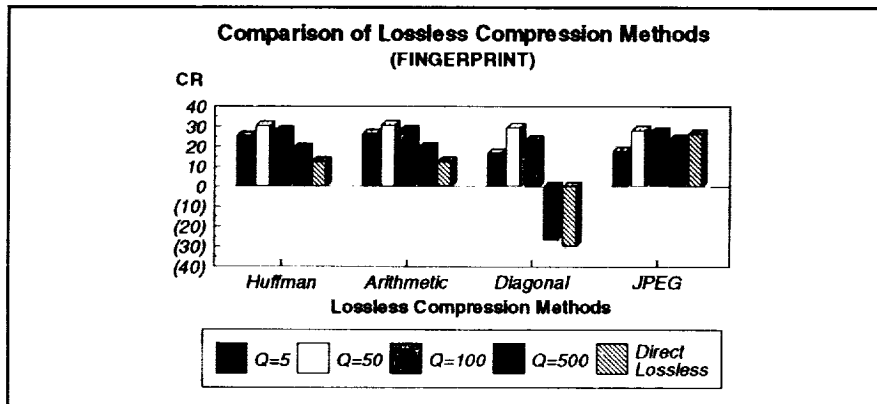


Figure 7c: Comparison of Hybrid Model with Lossless Compression Methods for FINGERPRINT at Various Quality Factors.

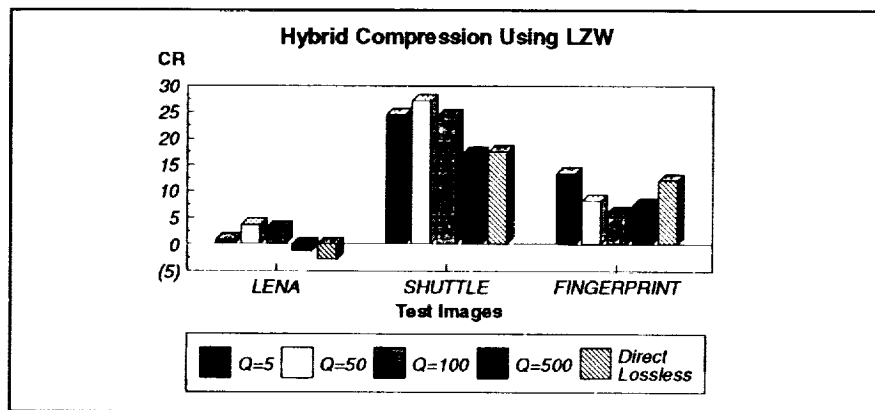


Figure 8: Compression Achieved Using LZW in Hybrid Model for Three Test Images at Various Quality Factors.

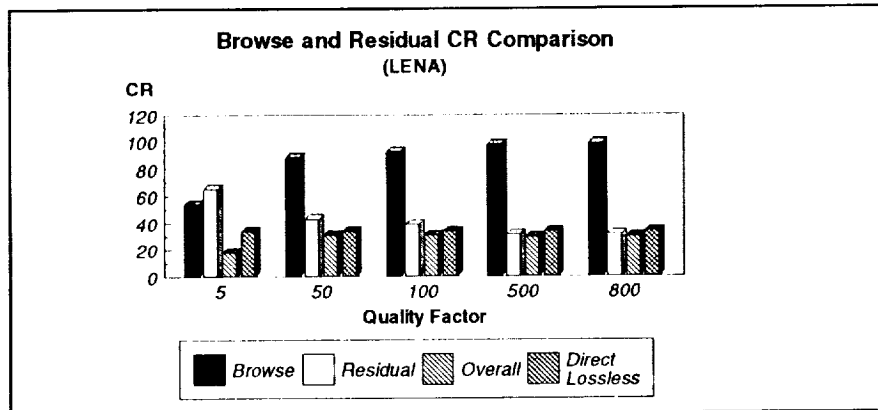


Figure 9: Browse and Residual CR Comparison with Direct Lossless Compression for LENA.

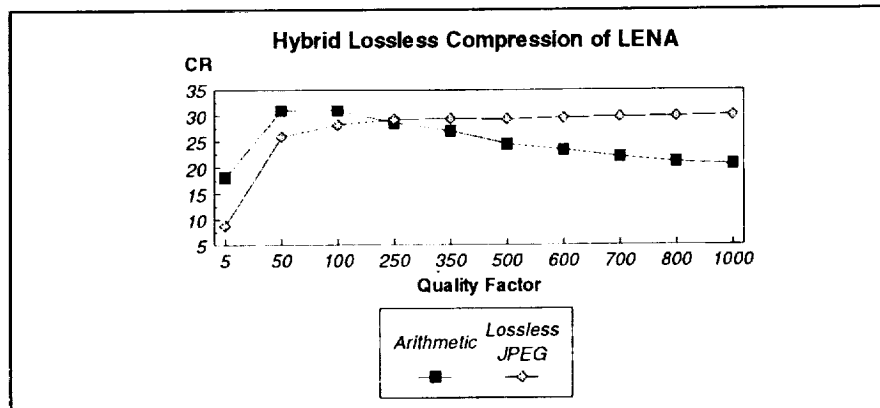


Figure 10a: Lossless Hybrid Compression of LENA Using Arithmetic and Lossless JPEG.

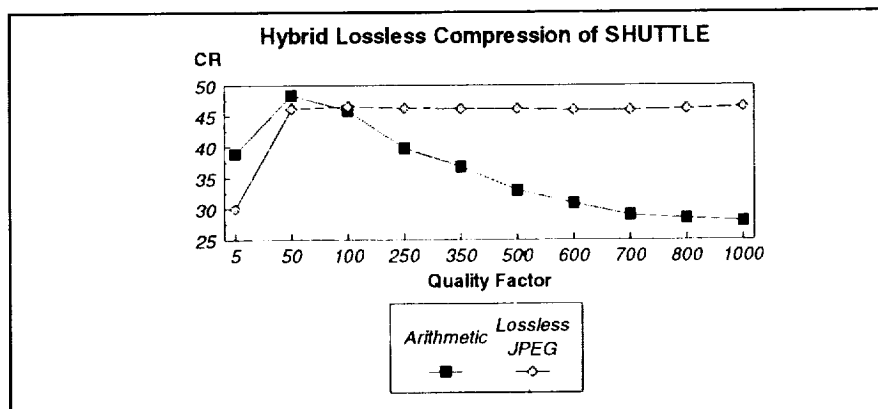


Figure 10b: Lossless Hybrid Compression of SHUTTLE Using Arithmetic and Lossless JPEG.

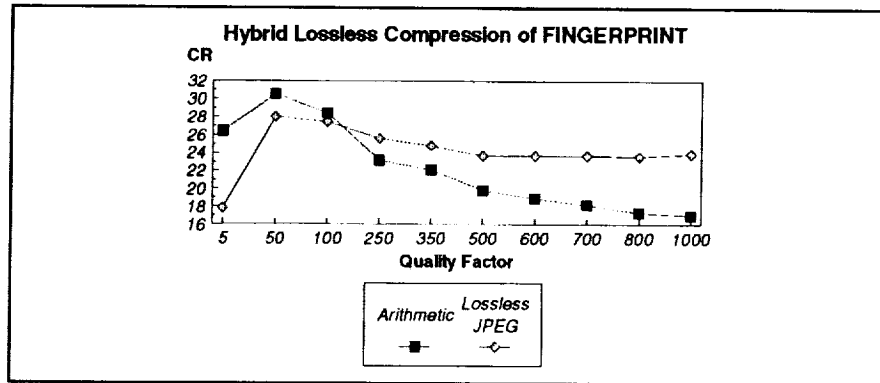


Figure 10c: Lossless Hybrid Compression of FINGERPRINT Using Arithmetic and Lossless JPEG.

6. Conclusions

Using the CR as a criterion for comparison, the results presented here indicate that the (lossy) JPEG DCT-based hybrid model has merit as a lossless image compression method. The results indicate that for low quality factors (≤ 50) arithmetic coding is the best choice for lossless compression of the residual images while at higher quality factors (> 50) lossless JPEG is the best choice. With the exception of lossless JPEG, the substitution of the other lossless compression methods (Huffman, arithmetic, LZW, and diagonal coding) into the hybrid model produce compression results that generally outperform their direct compression counterparts. CRs obtained for the lossless JPEG in the hybrid model were not predictably better than the CRs obtained by direct application of lossless JPEG. Nonetheless, the hybrid model has the advantage of decomposing the image into browse and residual components.

References

- [1] Novik, D.A., Tilton, J.C., and Manohar, M., "Compression Through Decomposition into Browse and Residual Images," *The 1993 Space and Earth Science Data Compression Workshop*, pp. 7-12, April, 1993.
- [2] Rabbani, M. and Jones, P.W., *Digital Image Compression Techniques*, SPIE Optical Engineering Press, Bellingham, WA, 1991.
- [3] Wallace, G.K., "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, v.38, pp. xviii-xxxiv, February, 1992.
- [4] Abbott, W.D., *A Simple, Low Overhead Data Compression Algorithm for Converting Lossy Compression Processes to Lossless*, Master's Thesis, Naval Postgraduate School, December, 1993.
- [5] Hung, A.C., PVRG-JPEG Codec 1.1, (Computer Program Documentation), Portable Video Research Group, Stanford University, August 1993.
- [6] Barnsley, M.F. and Hurd, P.H., *Fractal Image Compression*, AK Peters Ltd, Wellesey, MA, 1992.
- [7] Hannah, S.J., *A Hybrid Encoding Technique for Lossless Fractal Compression*, Master's Thesis, University of Alabama, September, 1993.
- [8] Nelson, M., *The Data Compression Book*, M&T Publishing, Inc., San Mateo, CA, 1992.
- [9] Gonzalez, R.C. and Wintz, P., *Digital Image Processing*, 2nd ed., Addison-Wesley Publishing Co., Inc., Reading, MA, 1987.

IMAGE QUALITY MEASURES AND THEIR PERFORMANCE

Ahmet M. Eskicioglu, Paul S. Fisher, and Siyuan Chen

University of North Texas

Department of Computer Sciences

P.O. Box 13886

Denton, TX 76203, USA

Telephone: (817) 565-2767

Fax: (817) 565-2799

E-mail: eskiciog@ponder.csci.unt.edu, fisher@gab.unt.edu,
schen@ponder.csci.unt.edu

56-61

460

p-13

Abstract

A number of quality measures are evaluated for gray scale image compression. They are all bivariate, exploiting the differences between corresponding pixels in the original and degraded images. It is shown that although some numerical measures correlate well with the observers' response for a given compression technique, they are not reliable for an evaluation across different techniques. The two graphical measures (histograms and Hosaka plots), however, can be used to appropriately specify not only the amount, but also the type of degradation in reconstructed images.

1. Introduction

The need for storing and transmitting huge volumes of data in today's computer and communications systems necessitates data compression in many fields ranging from medicine to aerospace. Data compression is an encoding process to reduce the storage and transmission requirements in applications. Many efficient techniques with considerably different features have recently been developed for both lossless and lossy compression. The evaluation of lossless techniques is normally a simple and straightforward task, where a number of standard criteria (compression ratio, execution time, etc.) are employed. A major problem in evaluating lossy techniques is the extreme difficulty in describing the type and amount of degradation in reconstructed images. Because of the inherent drawbacks associated with the subjective measures of image quality, there has been a great deal of interest in developing a quantitative measure, either in numerical or graphical form, that can consistently be used as a substitute. We would like to have such a measure not only to judge the quality of images obtained by a particular algorithm, but also for quality judgment across various algorithms. The latter task is definitely more challenging since a wide range of image impairments is involved. An extensive survey and a classification of the quality measures that appeared in the relevant literature are given in [1].

It is known that the mean square error (MSE), the most common objective criterion, or its variants do not correlate well with subjective quality measures. A major emphasis in recent research has therefore been given to a deeper analysis of the human visual system (HVS). The HVS is too complex to fully understand with present psychophysical means, but the incorporation of even a simplified model into objective measures reportedly leads to a better correlation with the response of the human observers.

We attempt to evaluate the usefulness of some of the objective quality measures listed in [1] through a set of experiments.

2. Image Quality Measures, Compression Techniques, and Test Images

The quality measures included in our evaluation are listed in Table 1. They are all discrete and bivariate, i.e., they provide some measure of closeness between two digital images by exploiting

the differences in the statistical distributions of pixel values. $F(j, k)$ and $\hat{F}(j, k)$ denote the samples of original and degraded image fields.

Table 1. Image Quality Measures

Average Difference	$AD = \sum_{j=1}^M \sum_{k=1}^N [F(j, k) - \hat{F}(j, k)] / MN$
Structural Content	$AC = \sum_{j=1}^M \sum_{k=1}^N [F(j, k)]^2 / \sum_{j=1}^M \sum_{k=1}^N [\hat{F}(j, k)]^2$
N. Cross-Correlation	$NK = \sum_{j=1}^M \sum_{k=1}^N F(j, k) \hat{F}(j, k) / \sum_{j=1}^M \sum_{k=1}^N [F(j, k)]^2$
Correlation Quality	$CQ = \sum_{j=1}^M \sum_{k=1}^N F(j, k) \hat{F}(j, k) / \sum_{j=1}^M \sum_{k=1}^N F(j, k)$
Maximum Difference	$MD = \text{Max}\{ F(j, k) - \hat{F}(j, k) \}$
Image Fidelity	$IF = 1 - (\sum_{j=1}^M \sum_{k=1}^N [F(j, k) - \hat{F}(j, k)]^2 / \sum_{j=1}^M \sum_{k=1}^N [F(j, k)]^2)$
Weighted Distance	WD: Every element of the difference matrix is normalized in some way and L_1 -norm is applied [1].
Laplacian Mean Square Error	$LMSE = \sum_{j=1}^{M-1} \sum_{k=2}^{N-1} [O\{F(j, k)\} - O\{\hat{F}(j, k)\}]^2 / \sum_{j=1}^{M-1} \sum_{k=2}^{N-1} [O\{F(j, k)\}]^2$
Peak Mean Square Error	$PMSE = \frac{1}{MN} \sum_{j=1}^M \sum_{k=1}^N [F(j, k) - \hat{F}(j, k)]^2 / [\text{Max}\{F(j, k)\}]^2$
N. Absolute Error	$NAE = \sum_{j=1}^M \sum_{k=1}^N O\{F(j, k)\} - O\{\hat{F}(j, k)\} / \sum_{j=1}^M \sum_{k=1}^N O\{F(j, k)\} $
N. Mean Square Error	$NMSE = \sum_{j=1}^M \sum_{k=1}^N [O\{F(j, k)\} - O\{\hat{F}(j, k)\}]^2 / \sum_{j=1}^M \sum_{k=1}^N [O\{F(j, k)\}]^2$
L_p -norm	$L_p = \{\frac{1}{MN} \sum_{j=1}^M \sum_{k=1}^N F(j, k) - \hat{F}(j, k) ^p\}^{1/p}, p = 1, 2, 3$
Hosaka plot	A graphical quality measure. The area and shape of the plot gives information about the type and amount of degradation [1,6].
Histogram	Another graphical quality measure. Gives the probability distribution of the pixel values in the difference image.
Note: For LMSE, $O\{F(j, k)\} = F(j+1, k) + F(j-1, k) + F(j, k+1) + F(j, k-1) - 4F(j, k)$. For NAE, NMSE, and L_2 -norm, $O\{F(j, k)\}$ is defined in three ways: (1) $O\{F(j, k)\} = F(j, k)$, (2) $O\{F(j, k)\} = F(j, k)^{1/3}$, (3) $O\{F(u, v)\} = H\{(u^2 + v^2)^{1/2}\} F(u, v)$ (in cosine transform domain).	

Among the few models of the HVS that have been developed, we chose the one proposed by Nill for dealing with cosine transforms. The function for the model is defined as [2]

$$H(r) = \begin{cases} 0.05r^{0.554}, & \text{for } r < 7 \\ e^{-9[\log_{10} r - \log_{10} 9]^{2.3}}, & \text{for } r \geq 7, \end{cases}$$

where $r = (u^2 + v^2)^{1/2}$, and u, v are the coordinates in the transform domain. The subimage structure weighting factor W_i in the original model was not used in our computations because we wanted to investigate the effect of $H(r)$ alone. Since W_i is proportional to the intensity level variance of subimage i , a separate analysis is needed to determine a suitable proportionality constant.

Table 2 Image Compression Techniques

JPEG	Fourth public release of the Independent JPEG Group's JPEG software
EPIC	Vision Science Group, The Media Laboratory, MIT
RLPQ	Department of Computer Sciences, University of North Texas
SLPQ	Department of Computer Sciences, University of North Texas

The implementations of the image compression techniques are given in Table 2. Both JPEG and EPIC belong to the class of transform coding techniques. The former performs the discrete cosine transform and the latter a wavelet transform. RLPQ and SLPQ contain several modifications to the Laplacian pyramidal decomposition and use a loose wavelet basis. After quantization, they employ arithmetic coding with a specifically tuned adaptive predictive model to compress the pyramid.

It should be noted that the choice of the compression techniques for an investigation of the performance of quality measures (especially those that are graphical) is important since it is desirable to include techniques which produce different types of impairments in the reconstructed images. Our purpose is to see how well the measures are able to describe image distortions of unsimilar nature. As we shall discuss later, the four codes in Table 2 serve this purpose.

The information about the three test images that we used can be seen in Table 3. Lenna and Fingerprint are in the set of the National Imagery Format Test Images. The third image, hurricane Gilbert, was obtained from the U.S. Navy.

Table 3 Test Images

Image	Source	Size(bytesxbytes)	Pixel Length(bits)	Spatial Frequency
Lenna	NITF	512x512	8	14.07
Gilbert	US Navy	512x512	8	31.25
Fingerprint	NITF	512x512	8	59.37

The spatial frequency for a given image is defined as follows [3]:

Consider an $M \times N$ image, where M = number of rows and N = number of columns. The row and column frequencies are given by

$$\text{Row_Freq} = \sqrt{\frac{1}{MN} \sum_{j=0}^{M-1} \sum_{k=1}^{N-1} [F(j,k) - F(j,k-1)]^2}$$

and

$$\text{Column_Freq} = \sqrt{\frac{1}{MN} \sum_{k=0}^{N-1} \sum_{j=1}^{M-1} [F(j,k) - F(j-1,k)]^2}$$

The total frequency is then

$$\text{Spatial frequency} = \sqrt{(\text{Row_Freq})^2 + (\text{Column_Freq})^2}.$$

This definition of frequency in the spatial domain indicates the overall activity level in an image.

3. Performance Of Quality Measures

The gray scale image data set was obtained by coding and decoding the three test images with the compression codes listed in Table 2. For each test image, seven different compression ratios were selected for degradation. They range from 10:1 to 70:1 with an increment of about 10. (Our original intention was to use the ratios 10:1, 20:1, 30:1, 40:1, 50:1, 60:1, and 70:1, but because of the inflexibility in using the JPEG parameter, we ended up with some different ratios.)

The photographic samples of the degraded images were first subjectively evaluated in an office environment by ten observers who were chosen from the graduate students and faculty having some background in image compression. They were asked to rank the images in two ways: Within each technique and between the four techniques for a fixed compression ratio. The mean rating of the group for an evaluation was computed by

$$R = \left(\sum_{k=1}^{10} s_k n_k \right) / \left(\sum_{k=1}^{10} n_k \right),$$

where s_k = the score corresponding to the k th rating, n_k = the number of observers with this rating, and 10 = the number of grades in the scale. No limits were imposed on viewing time or distance for the observers.

Table 4 shows the correlation between the numerical objective quality measures and the subjective evaluation. As a measure of the extent of the linear relationship, the Pearson's product-moment correlation coefficient (r) was used. The possible values of r are between -1 and +1; the closer r is to -1 or +1, the better the correlation is.

The coefficient values in Part (a) of Table 4 indicate that the quality measures can be put into three groups according to their performance:

- Group I: AD, SC
- Group II: NK, CQ, LMSE, MD
- Group III: WD, PMSE, IF, NAE, NMSE, L_p .

Table 4. (a) Correlation coefficients for each technique

(1) Lenna

Measure/Code	JPEG	EPIC	RLPQ	SLPQ
AD	0.528	-0.154	0.864	0.984
SC	0.561	-0.117	-0.988	-0.971
NK	0.479	0.865	0.996	0.979
CQ	0.480	0.865	0.996	0.979
LMSE	-0.980	-0.794	-0.752	-0.803
MD	-0.964	-0.984	-0.883	-0.941
WD	-0.995	-0.993	-0.954	-0.970
PMSE	-0.999	-0.996	-0.991	-0.990
IF	0.999	0.996	0.991	0.990
NAE	-0.997	-0.996	-0.970	-0.973
NAE(1/3)	-0.996	-0.996	-0.969	-0.972
NAE(HVS)	-0.972	-0.977	-0.925	-0.940
NMSE	-0.999	-0.996	-0.991	-0.990
NMSE(1/3)	-0.999	-0.997	-0.989	-0.989
NMSE(HVS)	-1.000	-0.998	-0.995	-0.996
L1	-0.997	-0.996	-0.970	-0.973
L2	-0.994	-0.993	-0.966	-0.969
L2(1/3)	-0.995	-0.993	-0.965	-0.968
L2(HVS)	-0.988	-0.990	-0.969	-0.975
L3	-0.991	-0.991	-0.961	-0.964

(2) Gilbert

Measure/Code	JPEG	EPIC	RLPQ	SLPQ
AD	0.747	-0.527	0.820	0.969
SC	-0.243	-0.936	-0.987	-0.930
NK	0.768	0.981	0.984	0.936
CQ	0.768	0.981	0.984	0.936
LMSE	-0.869	-0.800	-0.809	-0.727
MD	-0.828	-0.929	-0.853	-0.687
WD	-0.960	-0.960	-0.958	-0.923
PMSE	-0.979	-0.986	-0.981	-0.943
IF	0.979	0.986	0.981	0.943
NAE	-0.967	-0.975	-0.975	-0.939
NAE(1/3)	-0.842	-0.987	-0.974	-0.945
NAE(HVS)	-0.941	-0.941	-0.961	-0.914
NMSE	-0.979	-0.986	-0.981	-0.943
NMSE(1/3)	-0.717	-0.992	-0.978	-0.958
NMSE(HVS)	-0.988	-0.989	-0.998	-0.967
L1	-0.967	-0.975	-0.975	-0.939
L2	-0.961	-0.965	-0.962	-0.917
L2(1/3)	-0.754	-0.985	-0.959	-0.934
L2(HVS)	-0.964	-0.968	-0.985	-0.941
L3	-0.948	-0.960	-0.946	-0.890

(3) Fingerprint

Measure/Code	JPEG	EPIC	RLPQ	SLPQ
AD	0.803	-0.101	0.926	0.880
SC	0.325	-0.846	-0.955	-0.935
NK	0.895	0.975	0.958	0.944
CQ	0.895	0.975	0.958	0.944
LMSE	-0.906	-0.962	-0.737	-0.812
MD	-0.417	-0.956	-0.540	-0.402
WD	-0.962	-0.992	-0.938	-0.934
PMSE	-0.989	-0.999	-0.962	-0.953
IF	0.989	0.999	0.962	0.953
NAE	-0.975	-0.994	-0.956	-0.946
NAE(1/3)	-0.974	-0.993	-0.954	-0.939
NAE(HVS)	-0.948	-0.987	-0.936	-0.925
NMSE	-0.989	-0.999	-0.962	-0.953
NMSE(1/3)	-0.988	-0.995	-0.959	-0.934
NMSE(HVS)	-0.991	-0.996	-0.966	-0.954
L1	-0.975	-0.994	-0.956	-0.946
L2	-0.975	-0.995	-0.947	-0.937
L2(1/3)	-0.974	-0.993	-0.943	-0.920
L2(HVS)	-0.968	-0.997	-0.946	-0.930
L3	-0.975	-0.996	-0.934	-0.925

Table 4. (b) Correlation coefficients across techniques

(1) Lenna

Measure/Ratio	69:1	59:1	52:1	42:1	30:1	20:1	10:1
AD	-0.470	-0.498	-0.051	-0.558	0.875	0.260	-0.656
SC	0.863	0.716	0.863	0.626	0.683	-0.780	0.364
NK	-0.834	-0.705	-0.834	-0.675	-0.582	0.858	-0.455
CQ	-0.834	-0.705	-0.834	-0.675	-0.582	0.858	-0.455
LMSE	0.231	0.163	-0.010	0.203	-0.720	-0.471	0.950
MD	0.033	0.564	0.332	0.541	-0.380	-0.958	0.681
WD	-0.914	-0.221	-0.097	0.519	-0.254	-0.792	0.941
PMSE	0.188	0.533	0.360	0.671	-0.085	-0.893	0.929
IF	-0.161	-0.520	-0.349	-0.666	0.087	0.892	-0.928
NAE	-0.805	-0.295	-0.133	0.534	-0.015	-0.862	0.915
NAE(1/3)	-0.790	-0.417	-0.302	0.434	-0.017	-0.858	0.915
NAE(HVS)	0.454	0.527	0.270	0.531	-0.272	-0.828	0.874
NMSE	0.161	0.520	0.349	0.666	-0.087	-0.892	0.928
NMSE(1/3)	-0.627	-0.342	-0.349	0.384	-0.119	-0.879	0.928
NMSE(HVS)	0.589	0.664	0.397	0.629	-0.202	-0.879	0.909
L1	-0.805	-0.295	-0.133	0.534	-0.015	-0.862	0.915
L2	0.164	0.503	0.332	0.651	-0.086	-0.884	0.932
L2(1/3)	-0.607	-0.313	-0.326	0.370	-0.123	-0.867	0.934
L2(HVS)	0.553	0.632	0.373	0.604	-0.187	-0.864	0.894
L3	0.461	0.627	0.401	0.670	-0.139	-0.893	0.938

(2) Gilbert

Measure/Ratio	69:1	59:1	52:1	42:1	30:1	20:1	10:1
AD	-0.015	0.968	0.664	0.913	0.835	0.896	0.661
SC	-0.883	0.466	-0.494	-0.641	-0.552	-0.697	-0.739
NK	0.871	-0.654	0.617	0.728	0.636	0.760	0.741
CQ	0.871	-0.654	0.617	0.728	0.636	0.760	0.741
LMSE	0.532	-0.600	0.171	-0.112	-0.403	0.125	0.673
MD	-0.762	0.881	-0.935	-0.891	-0.761	-0.255	0.458
WD	-0.048	0.871	0.132	-0.365	-0.480	-0.639	-0.616
PMSE	-0.517	0.953	-0.700	-0.688	-0.788	-0.866	-0.753
IF	0.517	-0.953	0.700	0.688	0.788	0.866	0.753
NAE	-0.140	0.947	-0.011	-0.318	-0.374	-0.628	-0.759
NAE(1/3)	0.772	0.990	0.952	0.087	0.977	-0.174	-0.007
NAE(HVS)	-0.941	-0.961	-0.962	-0.896	-0.834	-0.854	-0.835
NMSE	-0.517	0.953	-0.700	-0.688	-0.788	-0.866	-0.753
NMSE(1/3)	0.560	0.993	0.961	0.118	0.982	-0.076	0.071
NMSE(HVS)	-0.967	-0.952	-0.973	-0.908	-0.843	-0.885	-0.895
L1	-0.140	0.947	-0.011	-0.318	-0.373	-0.628	-0.759
L2	-0.539	0.954	-0.712	-0.693	-0.786	-0.868	-0.754
L2(1/3)	0.584	0.999	0.935	0.084	0.974	-0.110	0.057
L2(HVS)	-0.965	-0.950	-0.967	-0.896	-0.832	-0.878	-0.881
L3	-0.787	0.984	-0.918	-0.904	-0.941	-0.893	-0.391

(3) Fingerprint

Measure/Ratio	69:1	59:1	52:1	42:1	30:1	20:1	10:1
AD	-0.871	0.878	-0.930	0.135	0.345	-0.093	-0.656
SC	-0.946	-0.925	-0.975	-0.960	-0.903	-0.953	-0.887
NK	0.979	0.930	0.982	0.971	0.924	0.966	0.920
CQ	0.979	0.930	0.982	0.971	0.924	0.966	0.920
LMSE	0.804	-0.437	-0.592	0.208	0.014	0.002	0.232
MD	0.735	0.977	0.999	0.309	0.573	-0.412	0.574
WD	0.057	-0.126	-0.976	-0.881	-0.918	-0.993	-0.930
PMSE	-0.185	0.916	-0.920	-0.983	-0.981	-0.989	-0.966
IF	0.185	-0.916	0.920	0.983	0.981	0.989	0.966
NAE	-0.304	1.000	-0.970	-0.999	-0.992	-0.989	-0.964
NAE(1/3)	-0.553	-0.024	-0.913	-0.994	-0.982	-0.980	-0.974
NAE(HVS)	-0.888	-0.404	-0.959	-0.977	-0.986	-0.946	-0.866
NMSE	-0.185	0.916	-0.920	-0.983	-0.981	-0.989	-0.966
NMSE(1/3)	-0.826	-0.791	-0.923	-0.986	-0.969	-0.976	-0.968
NMSE(HVS)	-0.894	-0.442	-0.986	-0.983	-0.979	-0.961	-0.902
L1	-0.304	1.000	-0.970	-0.999	-0.992	-0.989	-0.964
L2	-0.192	0.914	-0.921	-0.984	-0.983	-0.990	-0.964
L2(1/3)	-0.830	-0.792	-0.926	-0.987	-0.972	-0.974	-0.967
L2(HVS)	-0.896	-0.440	-0.988	-0.985	-0.983	-0.962	-0.892
L3	-0.195	0.862	-0.544	-0.960	-0.960	-0.988	-0.974

The measures in Group I cannot be reliably used with all techniques as the sign of the correlation coefficient does not remain the same. Group II measures are consistent, but nevertheless have poor correlation with the observers' response for some of the techniques. Among the useful measures in Group III, NMSE(HVS) is the best one for all the test images. Except for a single

case, the incorporation of the HVS into NMSE makes the correlation slightly stronger. For the other two measures NAE and L2, however, there is no such improvement. (In fact, the visual model has an adverse effect on NAE.) The results reported in [4] and [5] support our conclusion that the HVS model does not always improve the correlation, and when it does, the gain is small. The nonlinear filter $(\cdot)^{1/3}$, on the other hand, seems to have a random behavior, but usually leads to a weaker correlation. As IF is defined in terms of NMSE, the results for these two measures are identical. It has been found that PMSE establishes the same relationship as well.

Part (b) of Table 4 is rather disappointing, and the information that can be extracted is limited. As the compression ratio is increased, the measures perform much poorer. This observation is not surprising because different techniques introduce different types of degradation into the reconstructed images. Since the metrics combine all the pixel differences between two given images into a single number, one cannot expect to know much about the annoyance experienced by the human observer. In our experiments, for instance, although JPEG was the code for which the errors were always the smallest, the observers found the tile effect very objectionable in Lenna, yet favored blockiness in the higher frequency images Gilbert and Fingerprint.

To the best of our knowledge, histograms and Hosaka plots are the only two image quality measures that are graphical. Before we evaluate their performance, a specification of the type of impairment caused by the techniques is needed. Because of space limitation, the results for only the first test image will be discussed here. Four degraded versions of Lenna for the highest compression ratio (69:1) are given in Figure 1. The original image is also included for a comparison. The major types of degradation in the images are blockiness with JPEG, blurriness with EPIC, both fuzziness and blockiness with RLPQ, and fuzziness with SLPQ (The term fuzziness is used in the sense of equal amount of blurriness over the entire image).

A histogram of the compression error is constructed by plotting the number of times a specific value occurs in the difference image versus the value itself. Typically, it looks like a Gaussian curve; the more it resembles a spike at $x=0$, the greater the fidelity of the reconstructed image. The seven histograms in Figure 2 were obtained using JPEG. They clearly depict the increase in the amount of blockiness as the compression ratio goes up. The concentration of low intensity pixels for the lowest ratio is gradually reduced and the distribution becomes more uniform. Our experience has shown that histograms may also be used to specify different types of degradation in images. In Figure 3, the histograms with low intensity pixel concentrations are associated with RLPQ and SLPQ, and they are in contrast with those corresponding to JPEG and EPIC. The uniform fuzziness over the entire image, it is understood, leads to a spiky histogram. Nevertheless, the similarity between the histograms in each pair makes it difficult to distinguish between the artifacts involved.

To construct a Hosaka plot, or an h-plot, we measure a number of features of the reconstructed image and compare these with the corresponding features in the original image [6]. The difference between the two feature vectors generates a vector error measure, which, unlike scalar quantities, allows for a description of not only the amount, but also the type of degradation. In the process, the original image is first segmented into blocks whose variance is less than some specified threshold. These blocks are then grouped together to form a number of classes which depend on the size of the blocks. Two features are computed for each class in both the original and the reconstructed images. One of them is related to the mean intensity values and the other is the mean standard deviation. The h-plot is constructed by plotting the errors in the corresponding features in polar coordinates. The radius denotes the feature error, the left and right half planes contain the vectors associated with standard deviations and means, respectively.

It is reported in [6] that when noise is added to an image, the area of the h-plot is proportional to the image quality, but the structure of the diagram depends on the type of distortion. If an image is



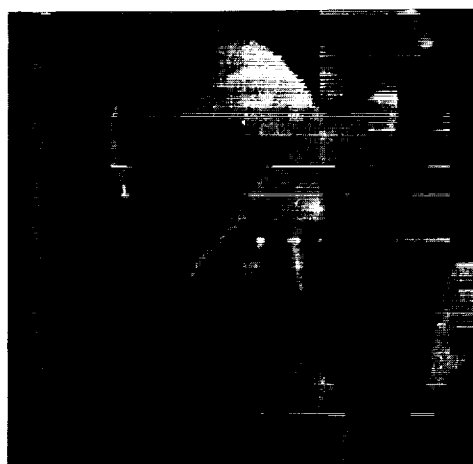
ORIGINAL



JPEG



EPIC



RLPQ



SLPQ

Figure 1

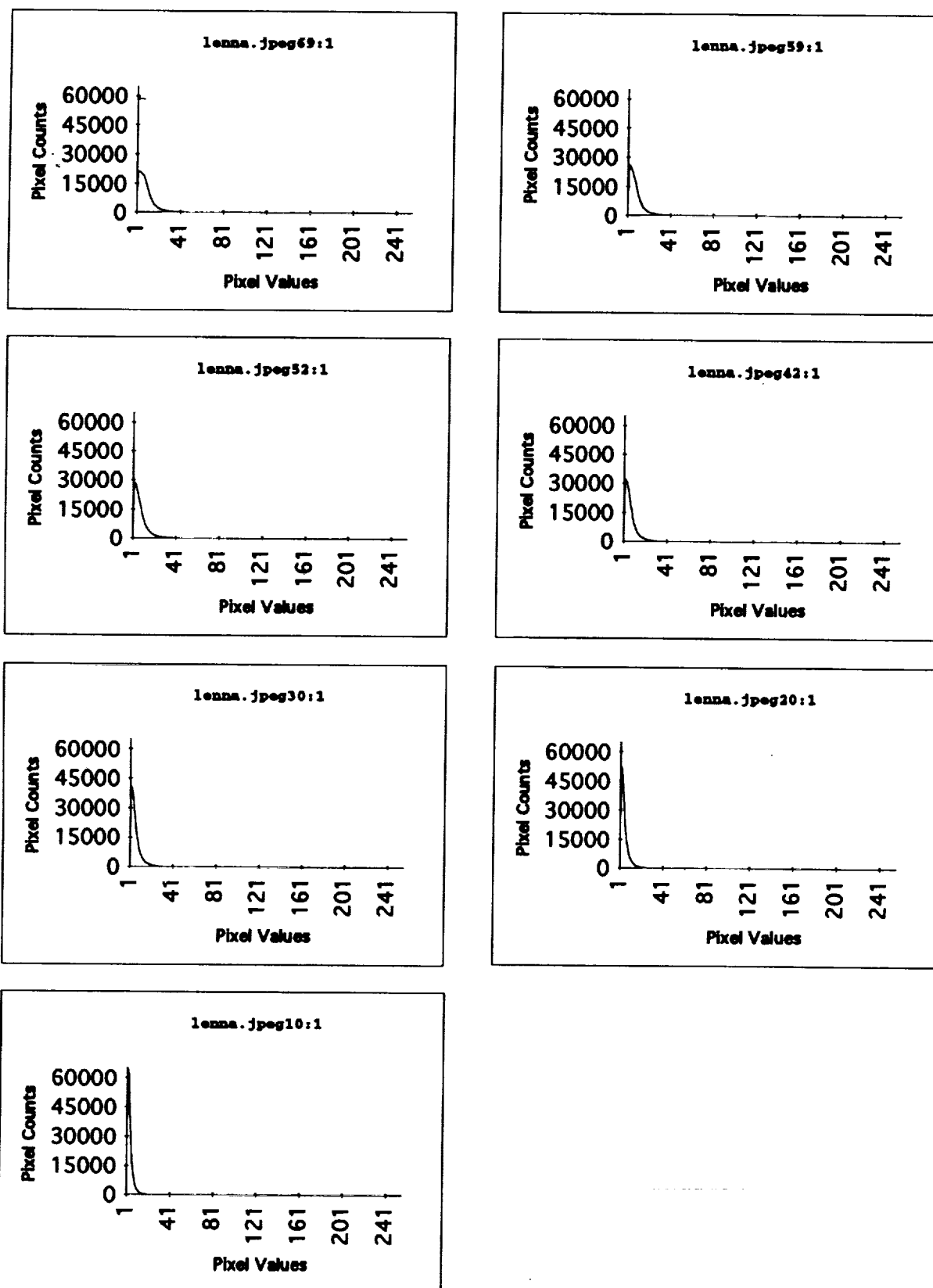


Figure 2

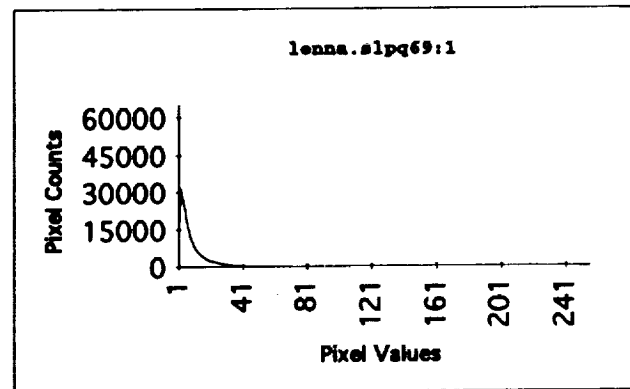
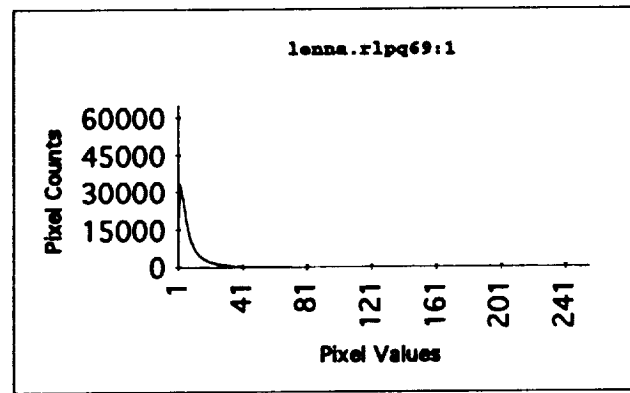
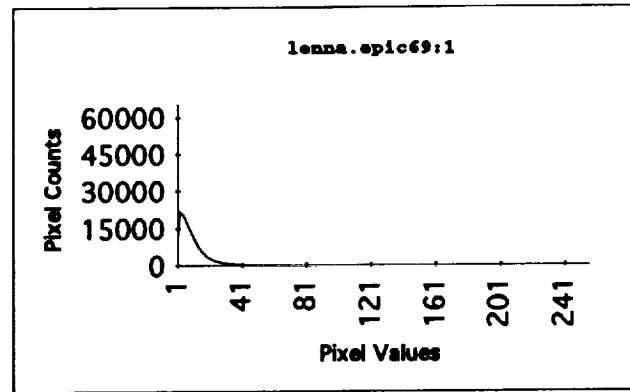
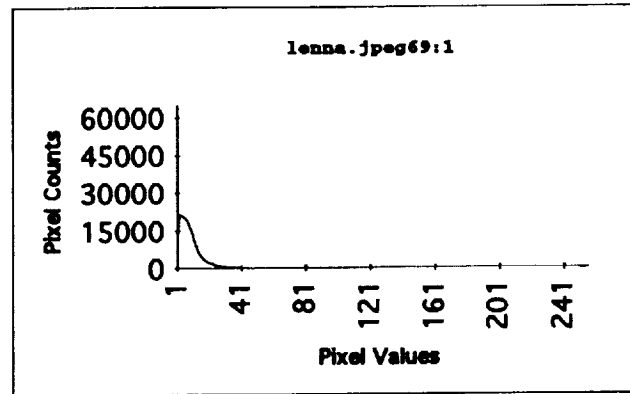


Figure 3

blurred, on the other hand, the pattern on the right side of the diagram remains fixed and increases in magnitude as the blurring increases while the left side is much less predictable.

The h-plots in Figure 4 were obtained using Lenna for all compression techniques and ratios. In each diagram, the length of a radius is 2.75 units. The blockiness is reflected on the right side of h-plots, whereas, the effect of blurriness can be traced on the left. By a simple comparison, we are able to see the way each code reduces the fidelity of the image. One can even learn how the distortion is distributed in the reconstructed images by looking at the relative lengths of the components along the axes. For example, it is evident that JPEG preserves the high frequency components (the feathers) of the image, whereas RLPQ induces uniform blockiness. Such information is extremely helpful considering the sensitivity of the human observer to the location of the image error. For the construction of the h-plots in Figure 4, the two parameters, the initial block size N and the variance threshold T , were chosen as 16 and 10, respectively, as in Hosaka's or Farrelle's work [6]. For high compression ratios, the h-plots for JPEG and RLPQ indicate that it may be worth trying larger values for these parameters.

4. Conclusions

The results of an evaluation concerning the usefulness of a number of objective quality measures for grayscale image compression have been presented. It is understood that although a group of numerical measures can reliably be used to specify the magnitude of degradation in reconstructed images for a given compression technique, an evaluation across different techniques is not possible. This is because a single scalar value cannot be used to describe a variety of impairments. A simple analogy would be the futility in comparing apples with oranges. The two graphical measures, however, are fairly successful in specifying the type of degradation. Hosaka plots, in particular, provide a good indication of how images are degraded. A combination of numerical and graphical measures may prove more useful in judging image quality. There is also a need for the development of new graphical measures with superior judgment capabilities. Further research in these areas is now ongoing.

References

- [1] A.M. Eskicioglu and Paul S. Fisher, "A Survey of Quality Measures for Gray Scale Image Compression," Proceedings of 1993 Space and Earth Science Data Compression Workshop (NASA Conference Publication 3191), Snowbird, Utah, April 2, 1993.
- [2] N.B. Nill, "A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment," *IEEE Transactions on Communications*, Vol. COM-33, No. 6, pp. 551-557, June 1985.
- [3] Paul S. Fisher and Nassrin Tavakoli, Final Report for Compression of Geophysical Data, Space and Naval Warfare Systems Command Contract N00039-C-0063, Department of Computer Sciences, University of North Texas, Texas, May 1992.
- [4] G.G. Kuperman and D.L. Wilson, "Objective and Subjective Assessment of Image Compression Algorithms," Society for Information Display International Symposium Digest of Technical Papers, Vol. 22, pp. 627-630, May 1991.
- [5] J. Farrell, H. Trontelj, C. Rosenberg, and J. Wiseman, "Perceptual Metrics for Monochrome Image Compression," Society for Information Display International Symposium Digest of Technical Papers, Vol. 22, pp. 631-634, 1991.
- [6] P.M. Farrelle, *Recursive Block Coding for Image Data Compression*, Springer-Verlag New York Inc., USA, 1990.

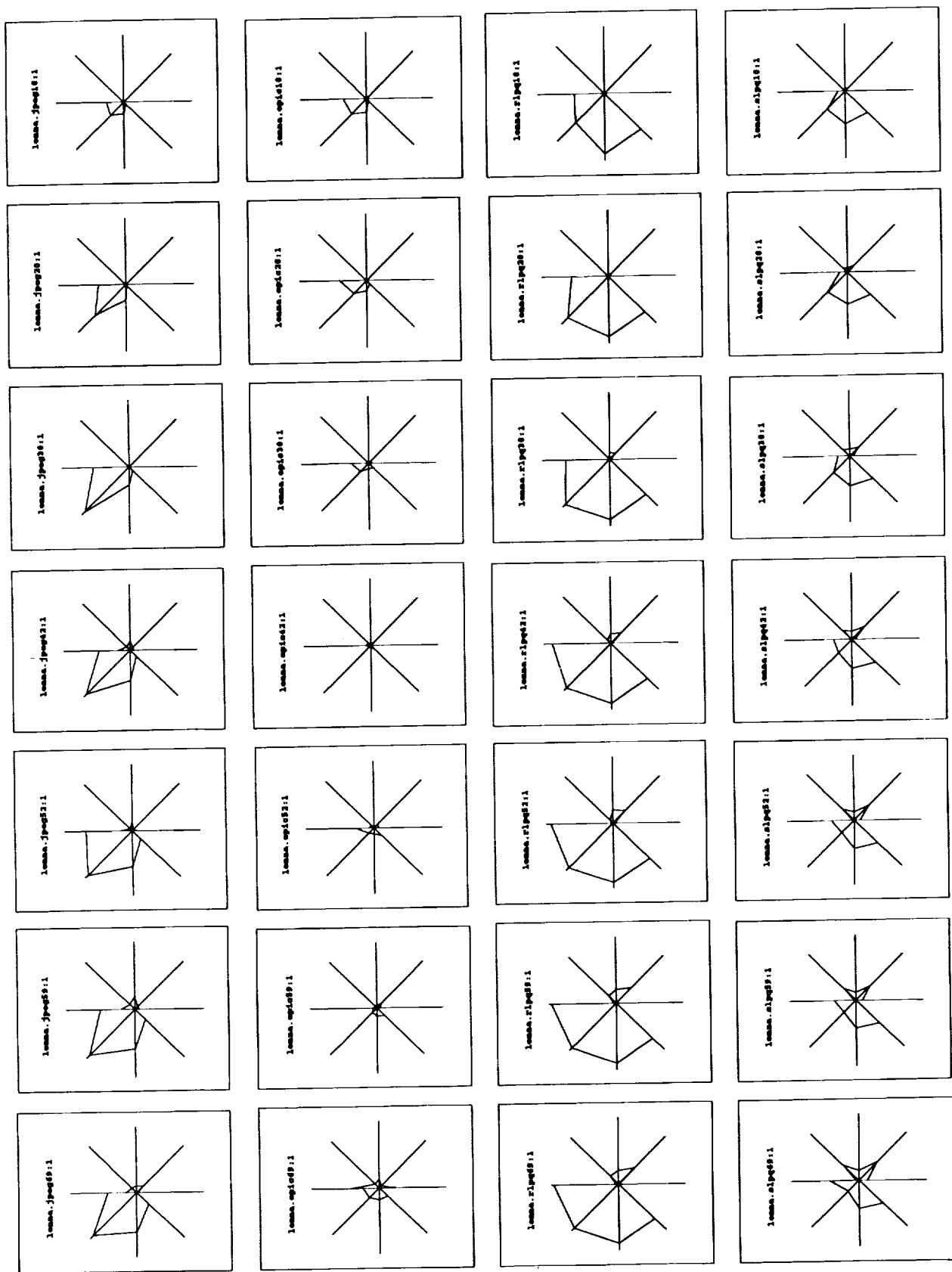


Figure 4

A Comparison Of Spectral Decorrelation Techniques And Performance Evaluation Metrics For A Wavelet-Based, Multispectral Data Compression Algorithm

Roy M. Matic
Hughes Research Laboratories
Malibu, CA
(310)317-5931
(310)317-5484 FAX
matic@maxwell.hrl.hac.com

Judith I. Mosley
Santa Barbara Research Center
Goleta, CA
(805) 562-7578
(805) 562-7881 FAX
5705@msgate.emis.hac.com

57-61

461

P-12

Abstract

Future space-based, remote sensing systems will have data transmission requirements that exceed available downlinks, necessitating the use of lossy compression techniques for multispectral data. In this paper, we describe several algorithms for lossy compression of multispectral data which combine spectral decorrelation techniques with an adaptive, wavelet-based, image compression algorithm to exploit both spectral and spatial correlation. We compare the performance of several different spectral decorrelation techniques, including wavelet transformation in the spectral dimension. The performance of each technique is evaluated at compression ratios ranging from 4:1 to 16:1. Performance measures used are visual examination, conventional distortion measures, and multispectral classification results. We also introduce a family of distortion metrics that are designed to quantify and predict the effect of compression artifacts on multispectral classification of the reconstructed data.

1. Introduction

In space-based, remote sensing systems, the limited ability to transmit sensor data to the ground places a major constraint on system feasibility. Available relay systems and direct downlink capabilities are not scaled to the data-transmission requirements for wide-area, high-resolution remote sensing systems envisioned for sensor systems of the year 2000 and beyond. Assuming data rates on the order of gigabits/sec for an advanced multispectral remote sensor system and a 600Mbps ATDRSS relay link, compression ratios on the order of 5-15:1 are required to transmit sensor output in real time. Since lossless compression techniques are not expected to achieve average compression ratios greater than 2.5:1, there is clearly a need to develop lossy compression techniques for multispectral data.

Previous work in the area of lossy multispectral compression has investigated a variety of different techniques. Baker and Tse¹ evaluated the performance of predictive coding, transform coding, and several vector quantization (VQ) techniques. In this work, only spectral correlations were exploited. The majority of other VQ techniques reported use VQ to exploit spatial correlations, and use predictive techniques (linear², nonlinear³, feature⁴, and polynomial⁵) to exploit spectral correlations. In the wavelet transform-based techniques that have been reported^{6,7}, a Karhunen-Loeve (KLT)⁸ transform or an approximation to it is performed prior to wavelet transformation to remove spectral redundancy in the data.

In this work, we use the wavelet transform in combination with several spectral decorrelation techniques to exploit both spectral and spatial correlation. Although the KLT is the optimum transform for the removal of spectral redundancy, it has historically been considered too computationally complex for real-time, on-board spacecraft implementation. In a previous paper⁹, we studied the performance of several prediction schemes to remove spectral redundancy. In this paper we examine the use of the wavelet transform to remove both spectral and spatial re-

dundancy. Both the prediction schemes and wavelet transform techniques are amenable to real-time implementation.

In addition, of greatest importance for multispectral remote sensing systems is the requirement that the compression process minimize the degradation of spatial and spectral fidelity to ensure that data exploitation is not compromised. Therefore, evaluation of lossy multispectral data compression techniques should include data exploitation simulations. However, comparison of exploitation performance is time consuming and is often impractical for compression algorithm development or parameter optimization. Conventional distortion measures (such as MSE or SNR) are not application sensitive and often do not accurately measure the effect of distortions on data exploitation. What is desired are quantitative degradation measures for exploitation algorithm performance characterization and prediction.

To address the need for meaningful image quality metrics, we introduce a set of metrics designed to quantify and predict the effect of compression artifacts on the performance of multispectral classification algorithms. These metrics, known as the Spectral Covariance Measures, are derived from the covariance matrices of the original, decompressed, and/or residual multispectral images. The goal of such metrics is to provide consistent predictive relationships between the quantitative distortion measure and a given application, such as Maximum Likelihood Multispectral Classification. Results are provided for the most promising of these measures, known as the Sum Delta Covariance Measure.

We simulate the performance of each compression algorithm on four multispectral (MS) images at compression ratios ranging from 4:1 to 16:1. An MS image consists of 8 co-registered 512x512 images, each representing a spectral band ranging from the Visible (Band 1) to the Near IR (Band 8). Performance measures used to evaluate the decompressed imagery are visual examination, conventional distortion measures (Mean Square Error), the Sum Delta Covariance Measure, and the results of Maximum Likelihood multispectral classification. We use these measures to determine the best spectral decorrelation technique, and to evaluate how well the Sum Delta Covariance Measure predicts multispectral classification performance.

The major contributions of this paper are simulation and performance evaluation of several different spectral decorrelation techniques, and preliminary results on the correlation between the Sum Delta Covariance Measure and Maximum Likelihood multispectral classification performance.

2. Compression Algorithm Description

A block diagram of the compression algorithms evaluated in this paper is shown in Fig. 1. The compression algorithms consist of a spectral decorrelation stage, a wavelet transformation stage, a rate allocation stage, a quantization stage, and an entropy coder stage. Each of these stages is described below.

2.1 Spectral Decorrelation Stage

We evaluated six different spectral decorrelation techniques: 1) Spatial-only (i.e., no spectral decorrelation), 2) Karhunen-Loeve transform (KLT), 3) Prediction with Two Reference Bands, 4) Band-to-band successive subtraction, 5) One dimensional wavelet transformation, and 6) Three dimensional wavelet transformation. In the Spatial-only technique, no spectral decorrelation is performed. Our purpose in evaluating this technique is to determine how much compression improvement (as measured by image quality and exploitability) can be obtained by

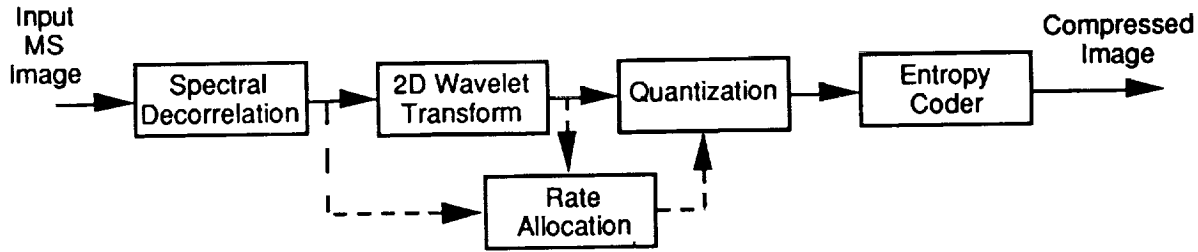


Fig. 1 Block Diagram of The Multispectral Compression Algorithms.

exploiting band-to-band spectral correlation. We also included the KLT in our evaluations so that its performance could be used as a reference to evaluate the performance of the other spectral decorrelation techniques.

Techniques 3 and 4 are differential schemes, in which the pixel values of a spectral band are replaced by the difference between the pixel values of the band and a predicted pixel value. In both schemes, the predicted value is obtained by using the value of a pixel at the same location, but in a different spectral band (known as the reference band). The motivation for these techniques is that because of spectral correlation, the predicted pixel value should be a reasonable estimate of the actual pixel value. The resulting differential band will have a lower entropy than the original band and will, therefore, be easier to compress. In the Two Reference Band approach (Technique 3), the predicted values for Bands 1, 2, and 4-6 are obtained by using the values of Band 3, and for Band 8, the predicted values are those of Band 7. In this technique, the values of the reference bands (Bands 3 and 7) are not changed. In the Successive subtraction approach (Technique 4), the reference band is just the next adjacent spectral band. For example, the reference band for Band 8 is Band 7, the reference band for Band 7 is Band 6, etc.. In this technique, the pixel values of Band 1 are not changed. To improve the performance of these two techniques, a normalization is performed prior to subtraction: the mean of each band is subtracted and the band variances are made identical by multiplication by a scaling factor.

In Techniques 5 and 6, we use the wavelet transform as a spectral decorrelation technique. In Technique 5, we perform a one dimensional wavelet transform on each multispectral pixel, prior to performing a two dimensional wavelet transform on each decorrelated band. In Technique 6, we perform a three dimensional wavelet transform to simultaneously remove both spectral and spatial redundancy. In both techniques, the wavelet filters used in the spectral dimension are the Haar (or Daubechies 1) filters. We use these filters because their implementation requires only two filter taps, which, with 8 spectral bands, permits a three level transform in the spectral dimension. As in the prediction schemes described above, prior to performing the wavelet transform, we subtract the mean of each spectral band and make the variances of the bands equal - in this case equal to the maximum variance of the bands.

All of the spectral decorrelation techniques mentioned above are reversible - the original pixel values can be obtained from the spectrally decorrelated values. With the possible exception of the KLT, these techniques are also amenable to real-time implementation since they involve relatively few computations per multispectral pixel.

2.2 Wavelet Transformation Stage

After the spectral decorrelation stage (except in Technique 6 above), we apply a two-dimensional discrete wavelet transform (DWT) to the decorrelated spectral bands to reduce

pixel-to-pixel spatial redundancy. The wavelet transform is a subband decomposition, in which a bank of bandpass filters splits an image into a number of separate, spatial frequency components, called subbands. The motivation for this decomposition is that the subbands can be encoded more efficiently than the original image. Typically, different bit rates and even different coding techniques are used for each subband to take advantage of the statistical properties of the subband and to control or shape the coding errors in an optimal fashion.

Wavelets are a recently developed class of subband filters in which the impulse response of the filters are orthogonal to one another and are all scaled versions of a single function known as the wavelet. The subbands produced by the transform have good redundancy removal properties, are orientation specific, and contain multiresolution information on both the location and scale of features, particularly edges or discontinuities in the image¹⁰. The ability to efficiently represent image features (particularly edges) is one of the reasons that wavelet-based compression schemes provide reconstructed images with good visual quality. The 2D DWT used in this paper is equivalent to a pyramid subband decomposition, where the bandwidths of the subbands are related by powers of two and represent an octave-based frequency decomposition. The transform is implemented using two finite impulse response filters which are applied recursively to the lowest frequency subband¹⁰. In this paper, the 2D wavelet transformation stage consists of a 6-level DWT, using the Daubechies 9-7 biorthogonal, linear phase filters¹¹. Symmetric edge reflection is used to avoid the introduction of discontinuities due to image boundaries¹².

In our implementation of a three dimensional wavelet transform, we use the Haar filter in the spectral dimension and the Daubechies 9-7 biorthogonal filters in the spatial dimensions, with symmetric edge reflection at the data boundaries in all three dimensions. The 3D transform consists of 6 levels: 3 levels performed on all three dimensions, and 3 levels performed only on the spatial dimensions.

2.3 Rate Allocation Stage

The purpose of the rate allocation stage is to select the rate (in bits/coefficient) of the wavelet subbands so that the desired compression ratio is achieved with minimum distortion in the reconstructed images. The general approach is to allocate higher rates to subbands that contain more information. Subbands allocated higher rates will be quantized with less distortion or error (the difference between the coefficient value and its quantized value). In a previous paper⁹, we examined the performance of four different rate allocation techniques. In three of these techniques, rate allocation is performed in two stages. In the first stage which occurs after spectral decorrelation, rate is allocated among the decorrelated bands in the spatial domain. The decorrelated bands are then treated as separate, independent images in the second stage, which allocates rate among the different wavelet subbands. In the fourth technique, all of the spectral bands are treated as a single dataset and rate allocation is performed in a single stage after spectral decorrelation and wavelet transformation. Our simulation results indicate that the fourth approach has the best performance. Use of any of the two stage rate allocation techniques results in significantly poorer performance. Thus, we use the single stage technique exclusively in this analysis.

After spectral decorrelation and wavelet transformation, we determine a bit rate/subband using the following formula which allocates rate based on the variance of the subband:

$$R_i = \Theta + \frac{1}{2} \log_2 \left[\frac{\sigma_i^2}{\left(\prod_{k=1}^S \sigma_k^{2N_k} \right)^{1/N}} \right], \quad (1)$$

where R_i is the allocated rate for subband i , σ_i^2 is the variance of subband i , Θ is the desired average rate for the dataset, S is the number of subbands, N_k is the number of coefficients in subband k , and N is the total number of coefficients in the dataset (equal to the number of bands times the number of pixels in the band). For spectral decorrelation Techniques 1-5, the number of subbands S is equal to 152 (8 spectral bands times 19 subbands/spectral band) and for Technique 6, the number of subbands is 31.

Eq. 1 is the rate allocation formula found in [13] that we have modified to account for the different sizes of the wavelet subbands. One problem with this formula is that if the variance of a subband is too small compared to the geometric mean of all of the subband variances, then this formula will result in a negative rate for the subband. In this case, we remove from Eq. 1 those subbands allocated a negative rate in the previous calculation and recalculate the R_i . This process generally requires at most 2-3 iterations to converge. The subbands that have been removed are not coded. All of the coefficient values in these subbands are set to zero.

2.4 Quantization Stage

The quantization stage consists of two parts: stepsize selection and uniform quantization. The purpose of the stepsize selection process is to determine a quantizer stepsize for each subband so that the quantized subband will be entropy coded at the allocated bit rate. We use a search algorithm that iteratively selects a stepsize, quantizes the subband, and then measures the first order entropy of the quantized subband to determine if the quantized subband meets its allocated rate, which indicates the suitability of the selected quantizer stepsize. After a stepsize is selected for each subband, the wavelet coefficients of the subband are quantized by dividing the coefficient value by the stepsize and rounding to the nearest integer.

Currently the iterative search algorithm used to determine quantizer stepsize is too computationally intensive for real-time implementation. A future effort is to replace the iterative search algorithm with a table lookup approach, developed through training, that selects quantizer stepsize based on the desired rate and variance of the subband.

2.5 Entropy Coding Stage

In the entropy coding stage the quantized wavelet coefficients are mapped into a set of variable-length code words. More frequently used values are mapped to short length code words and less frequently used values to long code words. Compression is achieved because the average number of bits to represent the output codewords is less than the average number of bits used to represent the quantized wavelet coefficients.

Our entropy coder is a hybrid that combines two well known techniques: the Rice coder¹⁴ and an arithmetic coder¹⁵. We use these two techniques in a complementary fashion. The Rice coder works well on short sequences and on sequences that have a first order entropy greater than 2bits/symbol. The arithmetic coder works well on long sequences that have low first order entropies (i.e., < 2bits/symbol). In coding each subband, we select the technique based on the size of the subband and its allocated bit rate. Performance simulations of this hybrid entropy

coder demonstrate coding efficiencies within 5-10% of information theoretical performance (based on first order entropy), which is significantly better than the performance of either technique alone.

2.6 Algorithm Summary

In Table 1, we list the different compression algorithms evaluated in this paper. For each algorithm in the table, we indicate the spectral decorrelation technique that is used. We also assign to each technique a short alpha-numeric symbol that we use to identify the specific technique in the graphs and tables of this paper.

Algorithm Symbol	Spectral Decorrelation Technique
Spatial-only	None
KLT	Karhunen-Loeue Transform
PRED1	Two Reference Band Predictor
PRED2	Successive Subtraction Predictor
WV1D	1D Wavelet Transform
WV3D	3D Wavelet Transform

Table 1. Multispectral Compression Algorithms

3. Performance Measures and Methodology

The goal of the compression schemes studied in this paper is to achieve a desired compression ratio with minimum distortion in the reconstructed MS image. One of the most common criteria used to measure distortion is the Mean Square Error (MSE):

$$MSE = \frac{1}{NN_B} \sum_{i=1}^{N_s} \sum_{j=1}^N (X_{ij} - \hat{X}_{ij})^2, \quad (2)$$

where N is the number of pixels in the spectral band, N_B is the number of spectral bands in the dataset, X_{ij} is the original pixel value of pixel j in Band i and $\hat{X}_{ij}$ is the pixel value after compression and decompression. We also measure the MSE for individual spectral bands. To calculate the MSE/band, we use an equation similar to Eq. 2, except that the summation is only over the pixels in the band.

Another criteria that we use to evaluate performance is a visual comparison between the reconstructed and original spectral bands of the MS images. We also viewed error images of the individual bands to study the types of errors introduced. The error images are constructed by taking the difference between the original and the reconstructed image and then scaling the errors to be in the range of 0-255 for display.

3.1 Spectral Covariance Measures

As a parallel effort to compression algorithm development and evaluation, we are investigating application specific distortion metrics. The objective of such a metric is to provide a predictive mapping between metric value and the change in performance of specific data exploitation applications after any process which introduces distortion to data, such as lossy compression. If such a relationship can be identified consistently between the metric and the application, then it will only be necessary to compute the metric to predict how the distortion process will affect the application. Ideally, such a metric should be straightforward to calculate and is

particularly useful if it correlates well to several applications (albeit perhaps via different predictive relationships).

For multispectral applications we have developed and investigated a set of measures called Spectral Covariance Measures. These measures are derived from the spectral covariance matrices of the original, decompressed and/or residual images. Design of these metrics is motivated by the fact that spectral principal components are the basis of many spectral feature extractors and that spectral covariance describes the degree of linear correlation between bands. An additional motivation is that some common classifiers, such as Mahalanobis Distance and Maximum Likelihood Classifiers, explicitly rely on spectral covariance to perform classification. We have investigated whether predictive relationships exist between these metrics and Maximum Likelihood Classifier performance. The most promising of the metrics, with respect to Multispectral Classification, is called the Sum Delta Covariance (SDC) metric. The SDC metric is computed as follows:

$$SDC = \sum_{\substack{\text{bandpairs} \\ ij}} |\text{Cov_original}_{ij} - \text{Cov_compressed}_{ij}|, \quad (3)$$

where all covariances are normalized. In this work we compare how well MSE and the SDC measure predict multispectral classifier performance.

3.2 Multispectral Classification Methodology

The fourth criteria used to evaluate the performance of the compression algorithms is to compare how well the compressed/decompressed imagery can be classified compared to the original multispectral (MS) images. A signature database defines the statistical characteristics of the proposed classes and is generated via training with representative MS data. The signature database is subsequently used by the MS classifier in conjunction with a decision rule to classify MS pixels. In general training may be supervised or unsupervised. For this study, unsupervised training is performed, due to lack of available ground truth. Both training and MS Classification are performed within the ERDAS GIS (Geographic Information Systems) and Image Processing environment. Unsupervised training is performed by the ISODATA clustering algorithm, and actual MS classification is performed using a Maximum Likelihood Decision Rule. Visual examination and measured signature divergence are used to iteratively edit and merge signatures derived from the original training images, yielding the final signature database.

In general we would like to use as much training data as possible to develop the signature databases, however for this effort we have a limited set of calibrated, registered MS images representing the spectral bands of immediate interest (Visible to Near IR). Specifically, this analysis is based upon 4 calibrated, co-registered MS images: 2 from each of 2 MS bandsets. These datasets are referred to as Airfield 1, Airfield 2, Urban 1 and Urban 2. Thus two signature databases are required for this analysis - one for each bandset. Eight spectral bands from each image were used. For this initial work, all eight bands were used for MS Classification. Future tasks will identify band subsets best suited for specific classification schemes and perform compression and exploitability analysis on these selected band subsets.

Each original MS image contains approximately 1000 X 700 MS pixels. For compression analysis, a 512X512 MS subimage was extracted from each image. The original (1000 X 700) images were used for classifier training. Thus each MS bandset's signature database is derived from two 1000 X 700 MS images. The image calibration data is used to "radiance normal-

ize" the data prior to training, such that within an MS bandset, the mapping from digital count to radiance is consistent.

In order to evaluate the impact of several compression algorithms on MS classification, each of the normalized uncompressed 512X512 MS images is submitted to the MS Classifier, using the appropriate signature database. This classification is treated as "truth" and becomes the basis for comparing classification results after compression. Data compression is performed on imagery which has not been radiance normalized, because raw sensor data which is input to an on-board compressor is typically unnormalized. After reconstruction, the compressed MS image is normalized using the same calibration and normalization factors which have been applied to the corresponding uncompressed MS image. This data is then submitted to the MS Classifier, using the appropriate signature database. The number of correctly classified pixels after compression is computed, yielding the percent correct classification results. This is done for each compression algorithm, at each compression ratio, for each 512X512 image.

4. Simulation Results

4.1 Compression Algorithm Performance

In Fig. 2 we compare the performance of the different spectral decorrelation techniques. In these two graphs, we display Mean Square Error as a function of compression ratio. Fig. 2a contains the results for dataset Airfield 1 and Fig. 2b contains the results for dataset Urban 1. From both of these graphs, it is clear that the KLT spectral decorrelation technique results in the best (i.e., lowest MSE) performance. For both datasets, the performance of the Two Reference Band technique and the 1D wavelet technique are comparable and have performance close to that of the KLT technique. For the Airfield 1 dataset, the performance of the Successive Subtraction technique and 3D wavelet technique are comparable and are better than not exploiting spectral decorrelation (the Spatial-only approach). However, in the Urban 1 dataset, the Successive Subtraction technique is actually worse than the Spatial-only approach, while the 3D wavelet technique still results in better performance.

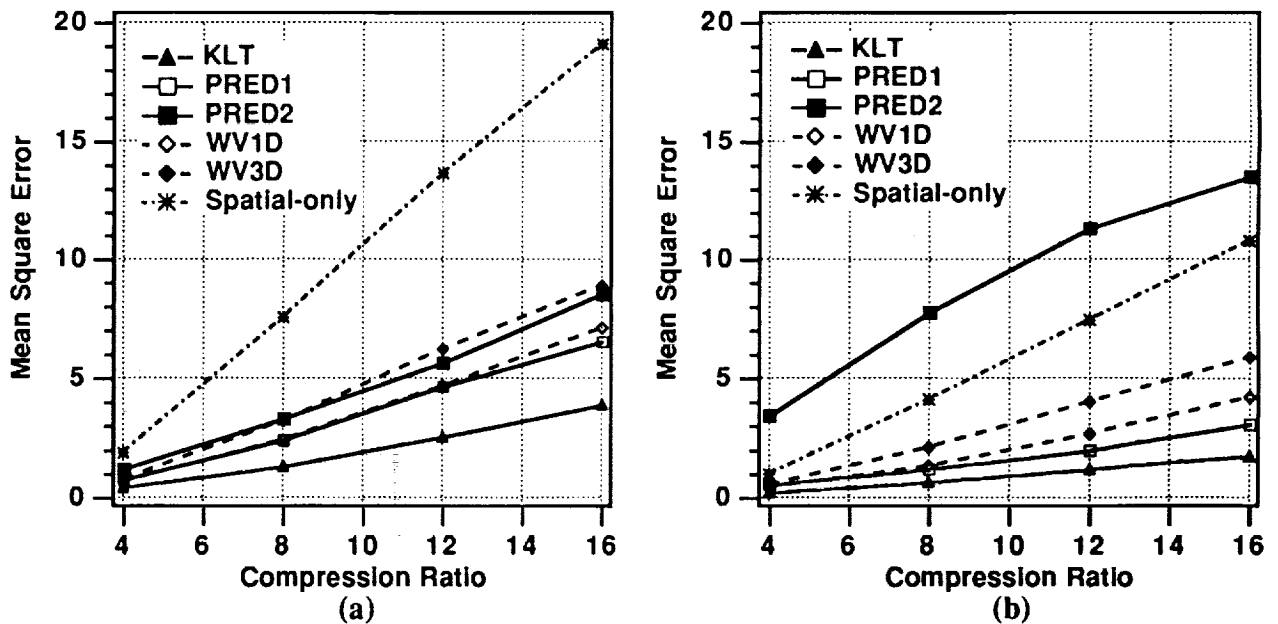


Fig. 2. Comparison of Different Spectral Decorrelation Techniques.
(a) Dataset Airfield 1 and (b) Dataset Urban 1.

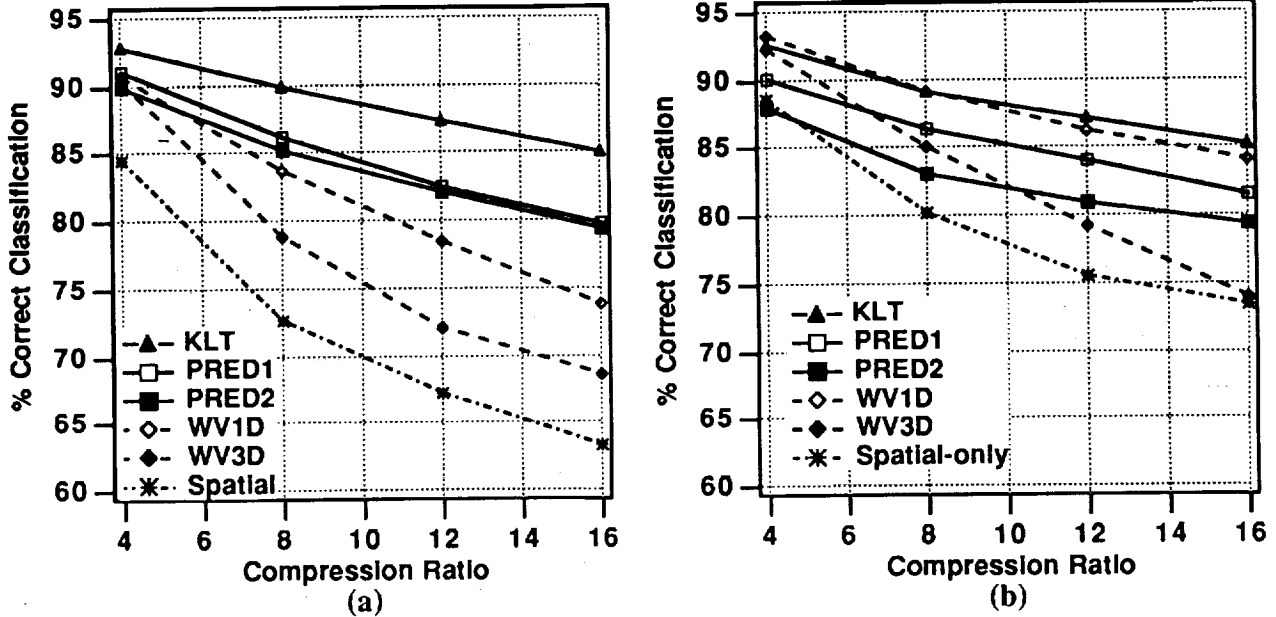


Fig. 3. Multispectral Classification Results.
(a) Dataset Airfield 1 and (b) Dataset Urban 1.

In Fig. 3 we show the results of performing multispectral classification on the reconstructed MS images. In both Figs. 3a and 3b, we display the percentage of MS pixels that are correctly classified as a function of compression ratio. As in Fig. 2, the KLT spectral decorrelation technique results in the best performance for both datasets. For dataset Airfield 1, the prediction schemes have similar classification performance, and both prediction techniques perform better than either the 1D or 3D wavelet-based techniques, which is a different relative performance ranking than the ranking obtained by comparing MSE performance. For dataset Urban 1, the classification performance of the 1D wavelet technique is almost as good as the KLT and significantly better than the prediction techniques or the 3D wavelet technique.

The relatively poor performance of the three dimensional wavelet transform approach may be due to the fact that there is a significantly smaller number of subbands (approximately a factor 5) in this approach than in any of the other approaches. The smaller number of subbands means that the subbands are larger than in the other approaches and, therefore, the bit rate allocation and quantization are more coarse. In other words, because the other techniques group the transform coefficients into a larger number of smaller groups, there is more flexibility in rate allocation and quantizer design. This additional flexibility translates into better performance.

4.2 Sum Delta Covariance vs. MSE Metric Performance Comparison

Because multispectral classification is applied to radiance normalized data, all MSE values used for metric evaluation are computed after radiance normalization of original and compressed imagery. Similarly, SDC is computed from radiance normalized data. Fig. 4 illustrates SDC vs. CR and MSE vs. CR for each compression algorithm for Airfield 1. When compared to Fig. 3a we see that neither SDC nor MSE consistently corresponds to the relative performance of the compression algorithms (as defined by classification accuracy).

In order to assess whether SDC shows promise as the basis of a predictive metric of classification accuracy, we have examined the correlation of both SDC and MSE to classification

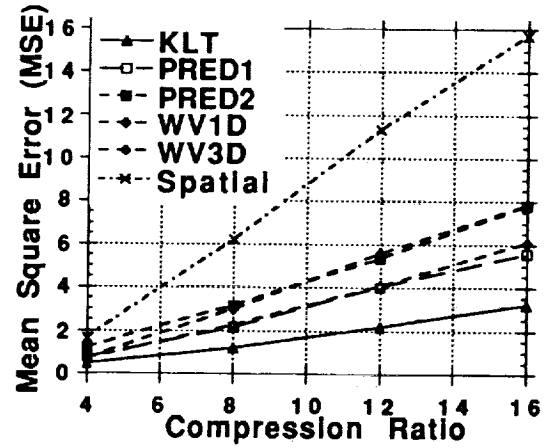
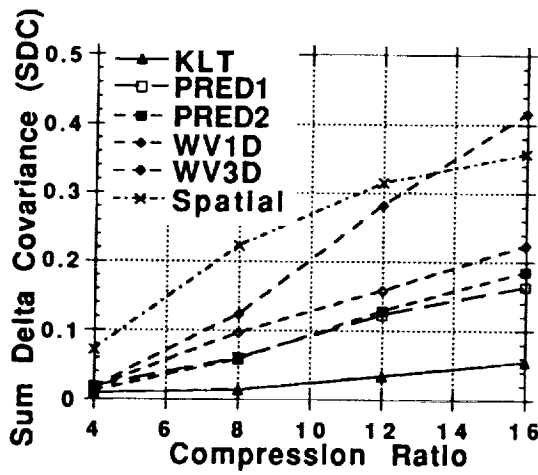


Fig. 4. (a) CR vs. SDC and (b) CR vs. MSE for Airfield 1

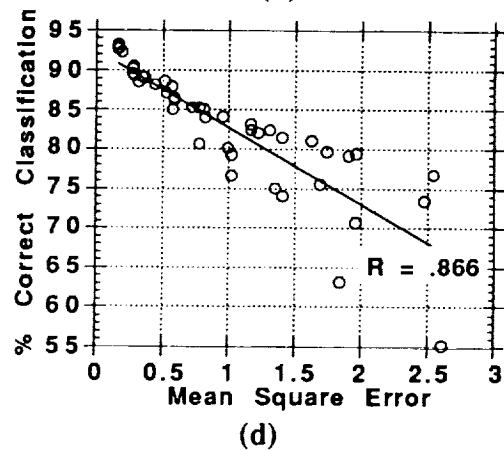
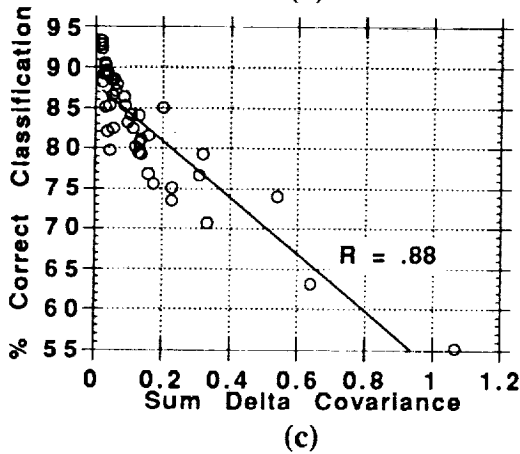
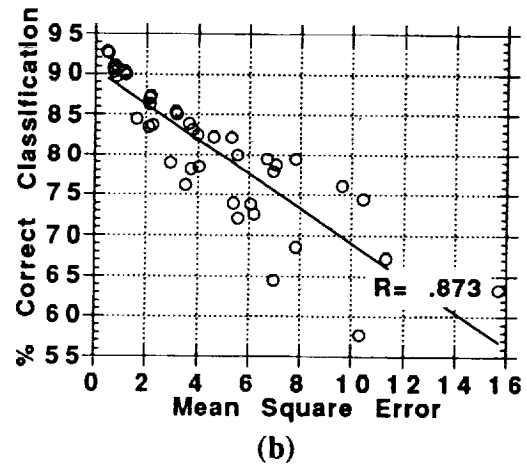
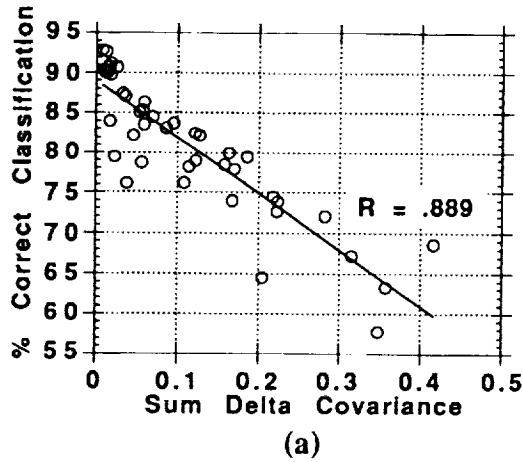


Fig.5. (a) SDC vs. Classification and (b) MSE vs. Classification for Airfield 1
(c) SDC vs. Classification and (d) MSE vs. Classification for Urban 1

accuracy. This is illustrated in Fig. 5 for each of the individual images. In these and the following figures, results are derived from 11 wavelet-based compression algorithms, including the six algorithms described in this paper and five algorithms described in a previous paper⁹. For any given image, SDC has only a slightly higher linear correlation to classification accuracy than

MSE. More important however, is what occurs when this correlation is examined over all images from both bandsets, as is illustrated in Fig. 6. When analyzed over both bandsets SDC has a notably higher correlation to classification accuracy than MSE. It appears that SDC is less sensitive than MSE to scene, sensor, and spectral variations. Thus it is possible that a refinement of the SDC measure will provide a useful predictive measure of classification accuracy.

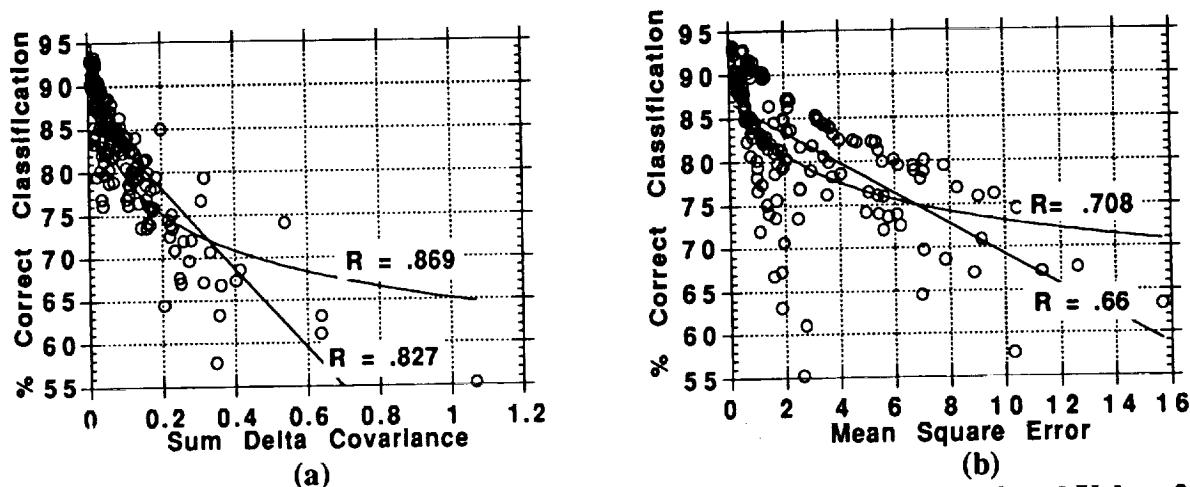


Fig. 6. (a) SDC vs. Classification for Airfield 1, Airfield 2, Urban 1 and Urban 2
(b) SDC vs. Classification for Airfield 1, Airfield 2, Urban 1 and Urban 2

5. Conclusions

In this paper, we have evaluated the performance of a number of wavelet-based multispectral compression algorithms. All of the algorithms use the wavelet transform to reduce pixel-to-pixel spatial redundancy. The difference in the compression algorithms lies in the techniques used to reduce band-to-band spectral correlation. Simulations of each of the compression algorithms was performed on four 8-band multispectral images at four different compression ratios. Visual examination, Mean Square Error, the Sum Delta Covariance Measure, and the results of multispectral classification of the decompressed images were the criteria used to evaluate the performance of the different algorithms.

As expected, the results of the simulations indicate that the Karhunen-Loeuvre transform is the best spectral decorrelation technique. Good performance is obtained with either a one dimensional spectral wavelet transform or a simple prediction scheme in which the pixel values of one of two bands is used to predict the pixel values in the remaining spectral bands. The performance of the three dimensional wavelet transform and that of the Successive subtraction prediction scheme were, in general, better than not exploiting spectral redundancy, but were significantly poorer than the other spectral decorrelation techniques.

We have implemented and evaluated a spectral covariance based metric called the Sum Delta Covariance. This metric correlated to multispectral classification accuracy more strongly than MSE and appears to be less sensitive than MSE to scene, sensor, and spectral variations. Thus this measure shows promise as the basis of a metric which can be used to predict multispectral classification accuracy.

In future directions of this research, we will concentrate on three areas: 1) development of improved compression algorithms, 2) an examination of sensor systems issues and their im-

pact on compression algorithm design and performance, and 3) development of improved compression evaluation techniques. Our focus in developing better compression algorithms is to evaluate different quantization schemes. For each processing stage we will tune algorithmic parameters and approaches for real-time on-board spacecraft implementation. Sensor systems issues that we plan to investigate are the effects on compression performance due to spectral band misregistration and detector nonuniformities. In the area of compression evaluation techniques, we plan to refine our classification techniques, the spectral covariance measures, and develop other application-specific image quality measures.

Acknowledgment

Special thanks to Gloria Yang for her support in performing the MS Classification analysis.

References

1. R. L. Baker and Y.T. Tse, "Compression of high spectral resolution imagery," in *Applications of Digital Image Processing XI*, Proc. SPIE 1974, 1988, pp. 255-264.
2. Y.T. Tse, S.Z. Kiang, C.Y. Chiu, and R.L. Baker, "Lossy compression techniques of hyperspectral imagery," *Proc. of Tenth Annual Intern. Geoscience and Remote Sensing Symposium (IGARSS'90)*, 1990, pp. 361-364.
3. S Gupta and A. Gersho, "Nonlinear predictive vector quantization of multispectral imagery," *Proc. Twenty-Fourth Asilomar Conference on Signals, Systems, and Computers*, vol. 1, 1990, pp. 331-335.
4. S. Gupta and A. Gersho, "Feature predictive vector quantization of multispectral images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. GEO-30, 1992, pp. 491-501.
5. D.D. Giusto, "On the compression of multispectral images," *Proc. of Intern. Geoscience and Remote Sensing Symposium*, 1990, pp. 1651-1654.
6. B.R. Epstein, R. Hingorani, J.M. Shapiro, and M. Czigler, "Multispectral KLT-wavelet data compression for Landsat Thematic Mapper images," *Proc. of Data Compression Conference (DCC'92)*, 1992, pp. 200-208.
7. T. Markas and J. Reif, "Multispectral image compression algorithms," *Proc. of Data Compression Conference (DCC'93)*, 1993, pp. 391-400.
8. W.K. Pratt, *Digital Image Processing*, John Wiley & Sons, 1978, pp. 259-264.
9. R.M. Matic and J.I. Mosley, "Wavelet transform - adaptive scalar quantization of multispectral data," *Proceedings of the Computing in Aerospace 9 Conference*, October 19-21, 1993.
10. O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, Oct. 1991, pp. 14-38.
11. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Proc.*, vol. 1, 1992, pp. 205-220.
12. G. Karlsson and M. Vetterli, "Extension of finite length signals for subband coding," *Signal Processing*, no. 17, 1989, pp. 161-168.
13. J.W. Woods, "Subband coding of images," *IEEE Trans. Acous. Speech Signal Proc.*, ASSP-34, 1986, pp. 1278-1288.
14. R.F. Rice, P. Yeh, W.H. Miller, "Algorithms for a very high speed universal noiseless coding module," *JPL Publication 91-1*, Jet Propulsion Laboratory, 1991.
15. I.H. Witten, R.M. Neal, and J.G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, 1987, pp. 520-540.

58-61
462

A Comparative Study of SAR Data Compression Schemes

N94- 28259

Dr. C.Lambert-Nebout

Dr. O.Besson

Dr. D.Massonnet, Dr. B. Rogron

CNES CT/AE/SEA/TB

18 av Edouard Belin

31055 Toulouse

e_mail : lambert@hathor.cnes.fr

phone : 33.61.27.33.08

fax : 33.61.28.19.96

ENSICA

49 av Léon Blum

31056 Toulouse

CNES CT/IA/QTIS/PR

18 av Edouard Belin

31055 Toulouse

I) INTRODUCTION

In spaceborne remote sensing, the amount of data collected has substantially increased in the last years. In the same time, the ability to store or transmit it has not increased as fast, so that there is a growing interest in developing compression schemes that could provide both higher compression ratios and lower encoding/decoding errors. In the case of the spaceborne Synthetic Aperture Radar (SAR) earth observation system developed by the French Space Agency (CNES), the volume of data to be processed is planned to exceed on-board storage capacities or telecommunication link. The objective of this paper is twofold:

- to present various compression schemes adapted to SAR data
- to define a set of evaluation criteria and compare the algorithms on SAR

data.

In this paper, we review two classical methods of SAR data compression and propose novel approaches based on Fourier Transforms and spectrum coding.

II) DESCRIPTION OF ALGORITHMS

a) Block Adaptive Quantizer

The first algorithm presented in this paper is the Block Adaptive Quantizer (BAQ) which was first proposed for the Magellan mission to Venus ([1]). This method encodes data into 2 bits in the following way: one bit is the sign bit, the other indicates the signal level. The signal-level bit indicates whether the signal is above or below a rms dependant threshold S :

$$\begin{aligned}x(n) &= "11" \text{ if } x < S \\x(n) &= "10" \text{ if } x \in [-S, 0] \\x(n) &= "00" \text{ if } x \in [0, S] \\x(n) &= "01" \text{ if } x > S\end{aligned}$$

In the decoding process, the signal $y(n)$ is reconstructed as follows:

$$\begin{aligned} y(n) &= (\text{sign}). \alpha.S \quad \text{if magnitude bit}=0 \\ y(n) &= (\text{sign}). \beta.S \quad \text{if magnitude bit}=1 \end{aligned}$$

The parameters α, β, S are chosen so as to minimize the encoding-decoding error:

$$\varepsilon = \int_0^S (x - \alpha S)^2 \cdot p(x) dx + \int_S^\infty (x - \beta S)^2 \cdot p(x) dx$$

where $p(x)$ is the probability density function (pdf) of the data. In the case of SAR data, one can assume a normal distribution $N(0, \sigma^2)$. By setting $S=k \sigma$, it can be shown that the optimal choice k_{opt} of k is given by the minimizer of the following function:

$$J(k) = \frac{1}{2} - \frac{(1 - e^{-k^2/2})^2}{\pi \cdot \text{erf}(k/\sqrt{2})} - \frac{e^{-k^2}}{\pi \cdot \text{erfc}(k/\sqrt{2})}$$

The optimal values of α, β are given by:

$$\begin{aligned} \alpha^{\text{opt}} &= \frac{\sqrt{2} \cdot (1 - e^{-k_{\text{opt}}^2/2})}{k_{\text{opt}} \cdot \sqrt{\pi} \cdot \text{erf}(k_{\text{opt}}/\sqrt{2})} \\ \beta^{\text{opt}} &= \frac{\sqrt{2} \cdot e^{-k_{\text{opt}}^2/2}}{k_{\text{opt}} \cdot \sqrt{\pi} \cdot \text{erfc}(k_{\text{opt}}/\sqrt{2})} \end{aligned}$$

Therefore, BAQ consists of the following steps:

- 1) select N samples
- 2) estimate σ from these samples
- 3) encode each sample as indicated above

The estimation of σ from the samples is not a direct estimation: it uses a mapping from the rms value to the average magnitude of the data ([1]): this method avoids multiplications and is therefore more attractive from an on-board point of view.

b) Block Floating Point Quantizer

The BFPQ method was proposed originally by Joo and Held ([2]) for the Magellan mission. As for BAQ, BFPQ uses results on gaussian signals quantization: it is known ([3]) that for a k -bit uniform quantizer, there exists an optimal value α_k^{opt} that minimizes the quantization

and saturation noise. The principle of BFPQ is to adapt the rms level of data to this optimal value while decreasing the number of quantization bits. If $x(i)$ denotes the original m -bit quantized signal, the compressed signal $y(i)$ is obtained by a simple division:

$$y(i) = \frac{x(i)}{C}$$

The constant C is determined using the fact that:

- i) $y(i)$ should be quantized on k bits
- ii) the rms of y is optimal

Then, it is straightforward to show that C is given by:

$$C = \frac{2 \cdot \sigma_x}{(2^k - 1) \cdot \sigma_k^{\text{opt}}}$$

where σ_x is the rms level of input data. The BFPQ encoding scheme consists of the following steps:

- 1) acquire N samples $x(i)$ $i=1..N$
- 2) estimate σ_x
- 3) calculate C
- 4) divide the original data by C

There exists numerous versions of this algorithm that can simplify it:

- i) σ_x can be estimated either directly either using the mapping method
- ii) C is rounded to the nearest power of 2: this enables the division to become a simple bit shift

An interesting implementation of the algorithm is to establish a direct mapping of σ_x to C 's nearest power of 2. In this case, BFPQ can be resumed by:

- 1) acquire N samples
- 2) estimate the average magnitude
- 3) read in a table the corresponding value of the scaling factor

This version requires only simple operations on integers and can be directly implemented on board.

c) FFT

In this section, we propose a generalisation of the popular Discrete Cosine Transform method of image compression ([4]) to the case of SAR data. As a matter of fact, DCT concerns real data and can not be applied directly to SAR data, which, by definition, is complex. We then propose to replace the Discrete Cosine Transform by a 2D Fast Fourier Transform (FFT), the compression scheme being now modeled by the following figure:

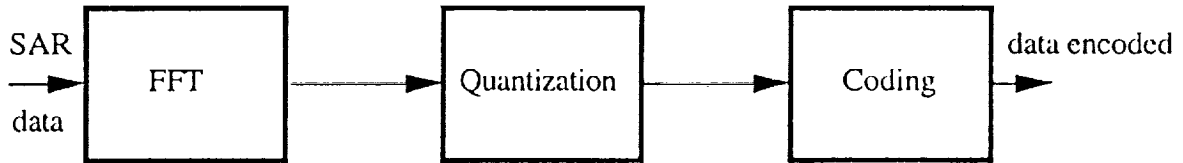


Figure 1: Block Diagram of FFT based SAR data compression

The original image is first partitioned into $N \times N$ pixel blocks and each block is independently transformed using the 2D Fourier Transform. The entropy of the transformed data is then estimated and the spectrum is quantized using 8 bits of resolution: given the original entropy, the quantization factor is chosen so that the entropy after quantization exactly matches the desired output bit rate. It is therefore supposed that the quantization process is optimal. Data is then coded using a lossless encoding algorithm (for instance, Huffman codes): since coding is supposed to be error free, it has not been simulated in this study. As can be seen, the algorithm used gives the optimal performance that can be achieved by this kind of method. It is to be noted that all the computations needed for this method were run using a floating point arithmetic, the analysis of errors due to fixed point implementation being beyond the scope of this study.

d) Presumming

The knowledge of some features of the radar signal suggests a more sensitive way to reduce the data flow in the spectral domain. In the range direction, the signal is shaped by the chirp generation which results in the spectral signature shown in figure 2:

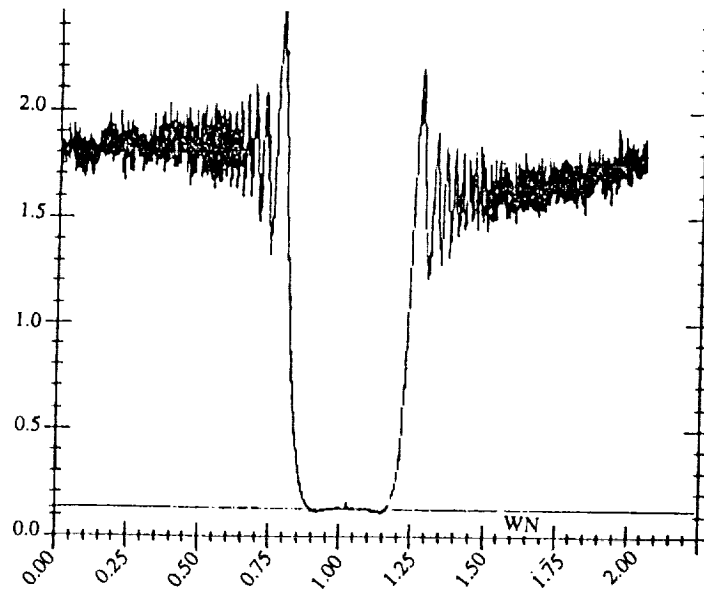


Figure 2: range spectral signature

The signal outside the "top hat" shape is noise and does not need encoding. In the azimuth direction, the signal is shaped by the antenna pattern once it has been aliased by the sampling phenomenon. Figure 3 shows an actual azimuth spectrum and, in dotted line, the actual shape of the antenna pattern once turned from the standard angular representation to the spectral representation. This spectral representation cannot be achieved in the real world due to insufficient pulse rate of the instrument. As a result, the outermost contribution of the antenna pattern is aliased in the actual spectrum. The signal can then be modelled into three parts :

- a white noise floor WN
- a useful radar signal RS
- a ambiguous radar signal AS

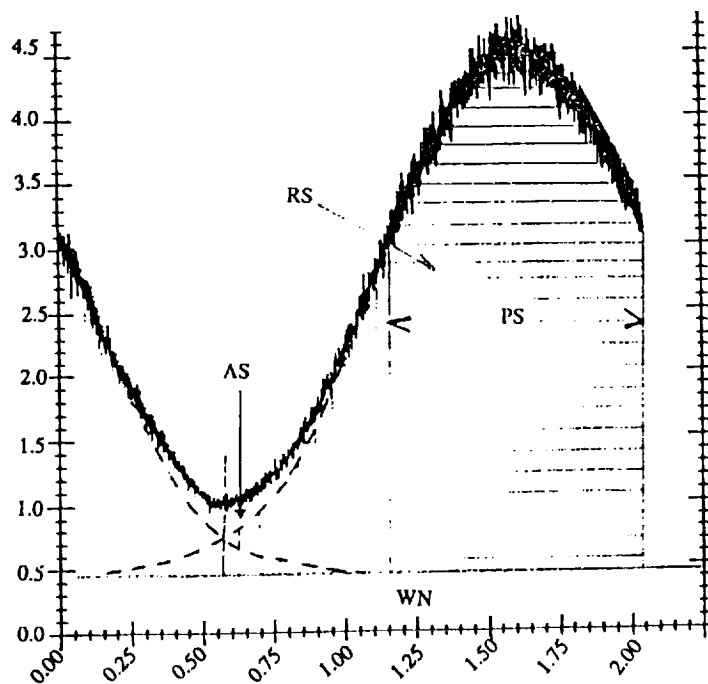


Figure 3: azimuth spectrum

The latter causes "ghosts" in the radar images, also called ambiguities, and should be eliminated. Standard compression schemes cannot make out a useful signal such as RS and an ambiguous signal AS since they have the same structure. It is also obvious that the signal to noise ratio is systematically greater in the central part of the spectrum.

The idea of presumming [5] is therefore to have a supervised coding of the 2D Fourier transform of the image. There would be no coding of the range region outside the useful signal (which results in a moderate saving of 20% or so). The coding in the azimuth spectrum would apply only to the central part where the signal to noise ratio is the highest. The loss of signal would amount to the vertically striped surfaces of figure 3, and the useful signal to the horizontally striped surface (the presummation span PS represented in figure 3 is just an illustration, not an actual value).

Presumming could easily achieve a factor of two in data compression with a minimal signal loss and an improvement of the quality due to the elimination of most of the ambiguous signal. This is true regardless of any further encoding of the conserved data.

III) EVALUATION CRITERIA

a) SAR data

In order to evaluate the performances of the different algorithms, a set of criteria were developped for both SAR data and SAR image. In the following, we suppose an image of width LX and height LY and note $z(i,j)$ (resp. $z'(i,j)$) the pixel of the i^{th} raw and the j^{th} column of the original (resp. encoded-decoded) data. The following criteria are considered for SAR data:

Mean Square Error (MSE):
$$\text{MSE} = \frac{1}{\text{LX} \cdot \text{LY}} \sum_{i=1}^{\text{LY}} \sum_{j=1}^{\text{LX}} |z(i,j) - z'(i,j)|^2$$

Maximum error:
$$E_{\max} = \max \left[\frac{|z(i,j) - z'(i,j)|}{|z(i,j)|} \right]$$

Phase Mean Square Error:
$$\text{MSE}_{\phi} = \frac{1}{\text{LX} \cdot \text{LY}} \sum_{i=1}^{\text{LY}} \sum_{j=1}^{\text{LX}} |\phi(i,j) - \phi'(i,j)|^2$$

Peak Signal to Quantization Noise Ratio:
$$\text{PSQNR} = 10 \cdot \log \left[\frac{\max(|z(i,j)|^2)}{\text{MSE}} \right]$$

Average Signal to Quantization Noise Ratio:
$$\text{ASQNR} = 10 \cdot \log \left[\frac{\frac{1}{\text{LX} \cdot \text{LY}} \sum_{i=1}^{\text{LY}} \sum_{j=1}^{\text{LX}} |z(i,j)|^2}{\text{MSE}} \right]$$

b) SAR image

An image acquired by ERS1 over southwest France in september 1991 was used as a testbed for the methods described in this paper. The image features ocean surface, homogeneous areas of forested or agricultural surfaces, highly contrasted areas such as the city of Bordeaux and some individual objects which are corner reflectors (two corner reflectors were placed in low backscatter regions) and which were shown to behave as corner reflectors (point targets).

A number of radar image quality criteria [6], which exceed the scope of this paper, were computed in addition to more standard data compression criteria, we may cite :

- range or azimuth resolution
- integrated sidelobe ratio
- ambiguous target ratio
- standard deviation/mean ratio over homogenous areas

More details about the results of this study are available in [7].

V) APPLICATION

The four above described methods were applied to an image provided from ERS1 and representing the scene of Cazaux (France). The original data had the following characteristics:

- * data precision: 5 bits per I and Q sample
- * data type: unsigned byte
- * data range: [0,31]
- * data entropy: approximately 4,7 bits
- * data properties: approximately Gaussian distributed with mean and rms:

$$m = 15.31866 + j 15.37417$$

$$\sigma = 6.733508 + j 6.706872$$

- * signal size: 10240 lines x 5616 complex samples

For all the methods, the image was partitioned into 128x128 blocks and each block was independently compressed and decompressed. The encoded/decoded data was then compared to original data by means of the above described criteria. The programs were written in Ansi C and run on a Sparc IPX station. The following tables show the SAR data evaluation criteria:

Criterion	BAQ	BFPQ(5,2)	FFT(5,2)	PRE(5,2)
mean	15.51 +j15.56	15.23 +j15.27	15.32 +j15.37	15.32 +j15.37
rms	6.94 +j6.92	4.67 +j4.64	6.86 +j6.83	6.4 +j6.386
MSE	9.758	19.625	7.044	12.8
E _{max}	6.4	1	9.487	14.56
MSE _φ	1.13E-1	1.17	6.65E-1	8.36E-1
PSQNR(dB)	17.2	14.165	18.615	16.02
ASQNR(dB)	9.7	6.67	11.12	8.53

Table I: SAR data evaluation criteria for 2 bit compression

Criterion	BFPQ(5,3)	FFT(5,3)	PRE(5,3)	PRE(5,3) no coding
mean	15.31 +j 15.35	15.32 +j 15.37	15.32 +j 15.37	15.32 +j 15.37
rms	5.7 +j 5.7	6.76 +j 6.74	5.23 +j 5.22	5.89 +j 5.88
MSE	5.28	1.85	9.957	19.048
$E_{\max}$	1	5.385	13.93	17.09
MSE_{ϕ}	7.11E-1	3.573E-1	7.41E-1	9.547E-1
PSQNR(dB)	19.87	24.42	17.11	14.294
ASQNR(dB)	12.37	16.92	9.62	6.8

Table II: SAR data evaluation criteria for 3 bit compression

Concerning SAR data, it seems that the FFT provides either for 2 or 3 bits the best results. Nevertheless, in the case of 2 bit compression, BAQ is shown to perform nearly as well as FFT: more, the computational requirements for BAQ are very inferior compared to FFT. Consequently, for a 2 bit compression scheme, BAQ seems to provide the best trade-off between performance and complexity. In the case of 3 bit compression, it is more difficult to establish a hierarchy between the methods: if FFT is shown to have the best performances, this algorithm is more complicated than BAQ, BFPQ and Presumming with no coding.

The major conclusions of SAR image criteria [7] could be itemized below :

- all algorithms produce errors on the phase of image pixel,
- FFT algorithm reproduces images better than the other algorithms,
- Presumming algorithm is a very interesting algorithm : its performance is very near to FFT (its complexity is lower),
- BFPQ (5,3) and BAQ (5,2) are however very similar to FFT in terms of image quality for a city.

The images before and after compression-decompression can be found at the end of the paper.

V) CONCLUSION

We have presented in this paper four compression algorithms for raw SAR data. These algorithms have been developed in C language on a SUN station. Their performances have been studied and compared through image quality criteria, data criteria and complexity criteria on data supplied from ERS-1. The choice of the best algorithm (specially for space on-board application) is indeed a trade-off between performance and complexity.

References:

- [1] R.Kwok,W.T.K.Johnson, "Block Adaptive Quantization of Magellan SAR Data", *IEEE Trans. Geo. Remote Sensing*, vol 27, n°4, pp 375-383, July 89
- [2] T.H.Joo,D.N.Held, "An Adaptive Quantization Method for Burst Mode Synthetic Aperture Radar Data", *IEEE Int. Radar Conf.*, pp 385-390, 1985
- [3] G.A.Gray,G.W.Zeoli, "Quantization and Saturation Noise Due to Analog-to-Digital Conversion", *IEEE Trans. Aerospace Electronic Systems*, pp 222-223, Jan 71
- [4] M.Rabbani,P.W.Jones, *Digital Image Compression Techniques*, Donald O'Shea Editors,1991
- [5] D. Massonnet , "Intérêt d'une Présommation embarquée sur un système SAR", *Internal memo CT/IA/QTIS/PR*, n°178 6-3-1992
- [6] D. Massonnet , "Radar Image Quality White Paper", *Proc. of the SAR calibration workshop Oberpfaffenhofen*, Germany, October 9th-11th 1991
- [7] B. Rogron , "Compression des données brutes SAR et Qualité des Images synthétisées", *Internal memo CT/IA/QTIS/PR* n°646 18-10-1993

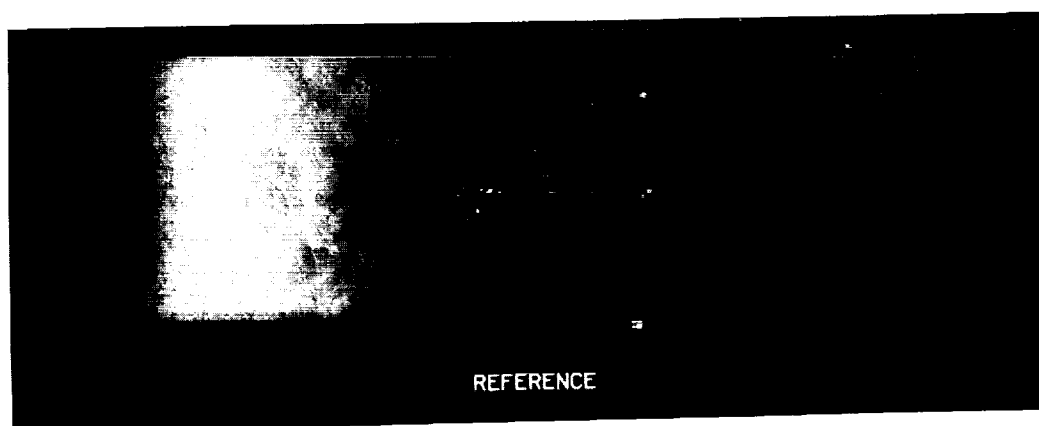


Figure 4: Reference image from ERS-1

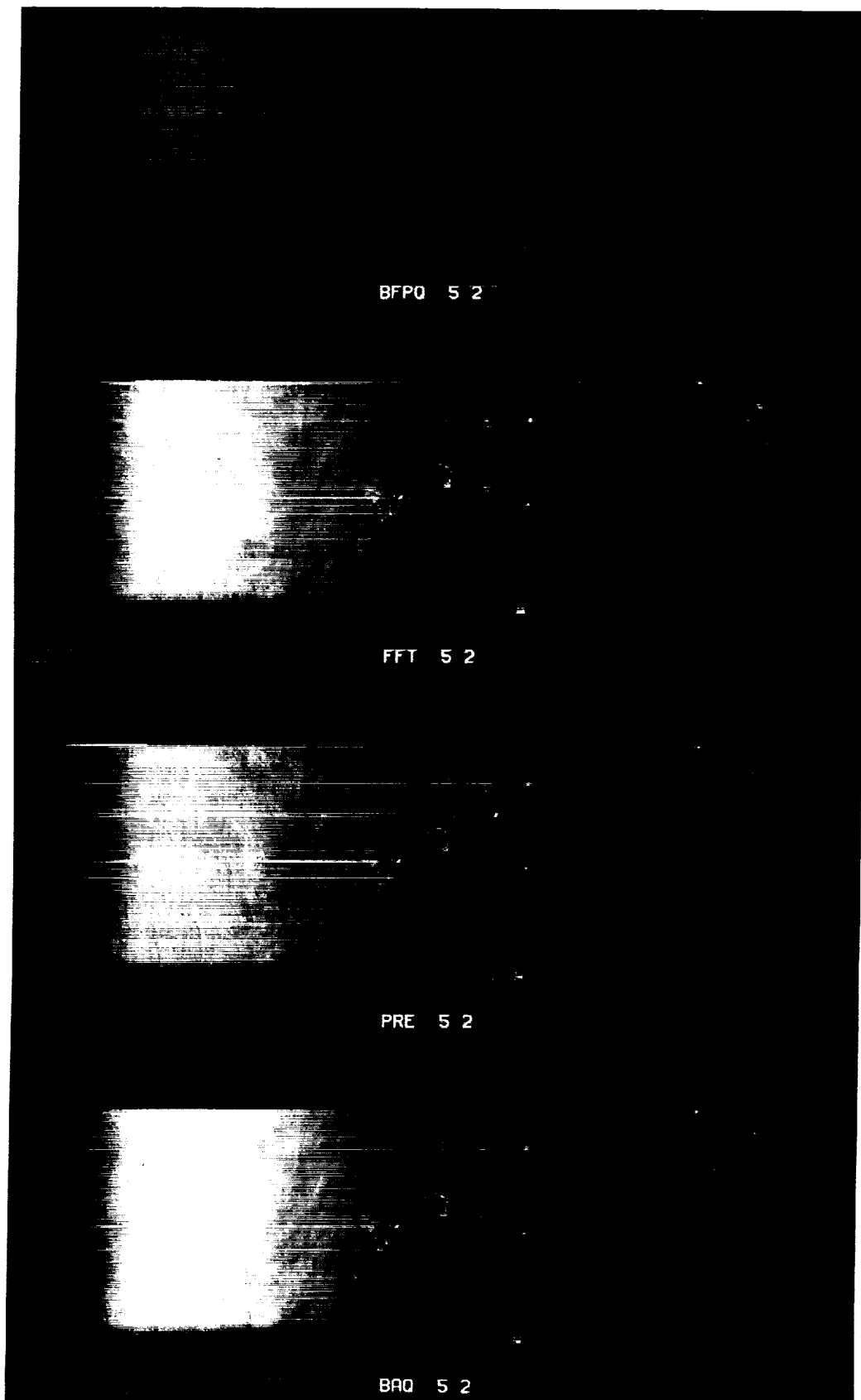


Figure 5: Image after 2 bit compression and decompression

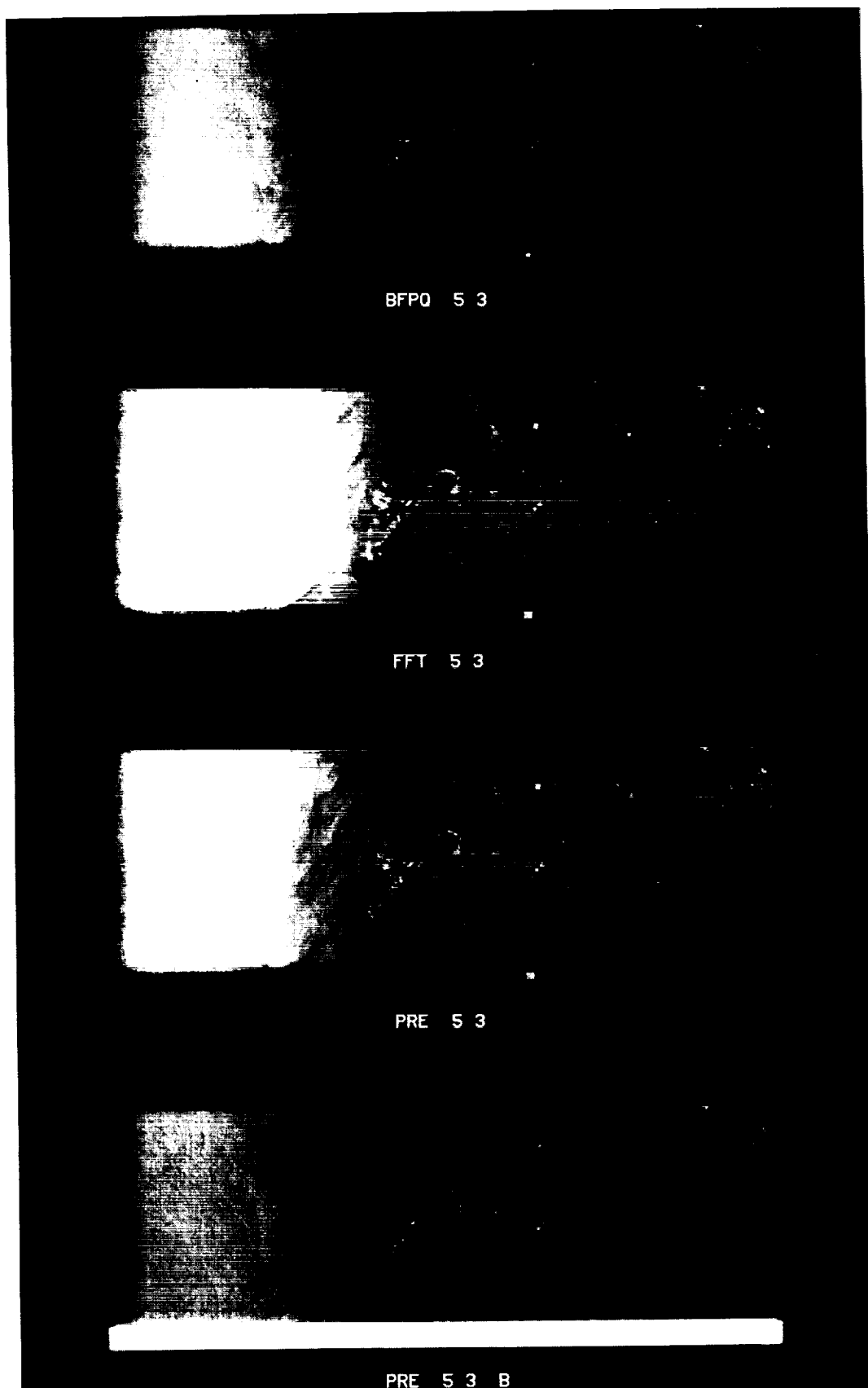


Figure 6: Image after 3 bit compression and decompression

7-61

463

p-11

Perceptual Compression of Magnitude-Detected Synthetic Aperture Radar Imagery¹

John D. Gorman and Susan A. Werness

Environmental Research Institute of Michigan, Box 134001, Ann Arbor, MI, 48113-4001
Voice: (313) 994-1200, FAX: (313) 994-5704, Email: gorman@erim.org, werness@erim.org

Abstract

We present a perceptually-based approach for compressing synthetic aperture radar (SAR) imagery. Key components of the approach are a multiresolution wavelet transform, a bit allocation mask based on an empirical human visual system (HVS) model, and hybrid scalar/vector quantization. Specifically, wavelet shrinkage techniques are used to segregate wavelet transform coefficients into three components: local means, edges, and texture. Each of these three components is then quantized separately according to a perceptually-based bit allocation scheme. Wavelet coefficients associated with local means and edges are quantized using high-rate scalar quantization while texture information is quantized using low-rate vector quantization.

We assess the impact of the perceptually-based multiresolution compression algorithm on visual image quality, impulse response, and texture properties for fine-resolution magnitude-detected SAR imagery and find excellent image quality at bit rates at or above 1 bpp along with graceful performance degradation at rates below 1 bpp.

1 Overview

We present a perceptually-based compression algorithm along with a preliminary evaluation of its performance on fine-resolution synthetic aperture radar (SAR) imagery. Properties of the algorithm are: (i) spatial adaptability to accommodate both the large dynamic ranges and unique image textures seen in SAR imagery, and (ii) the use of perceptually-based design criteria to optimize image quality rather than mean-squared error. Key components of the approach are a multiresolution wavelet transform, a bit allocation method based on an empirical human visual system (HVS) model, and hybrid scalar/vector quantization.

A consistent motivation for the multiresolution decomposition is its conceptual similarity to scene decompositions performed by the human visual system, which set the stage for application of simple, effective HVS bit allocation schemes. Our algorithm is similar in spirit to the wavelet coding techniques described in [1, 7, 11, 16] and the subband coding techniques in [14, 15]. The main distinction between our approach and others is the use of a fixed-weight perceptually-based bit allocation scheme that accounts for both the large dynamic range and texture patterns (speckle) present in SAR imagery.

Wavelet shrinkage techniques [6] are used to segregate wavelet transform coefficients into three components: local means, edges, and texture. Each of these three components is then quantized separately according to a perceptually-based bit allocation scheme. Because edges and low frequency information are perceptually most important [13], wavelet coefficients associated with local means and edges are quantized using high-rate scalar quantization

¹This work supported in part under internal ERIM funding.

while texture information is quantized using low-rate vector quantization. A minimum rate constraint is set for the local mean and edge components so that essential image content is preserved even at bit rates as low as 1/8 bpp.

The perceptually-based bit allocation scheme is implemented by applying a bit-allocation weighting table to the wavelet transform coefficients. Our approach uses a fixed table rather than the weighted mean-squared error approach reported in [14]; in the latter reference, a data-dependent bit allocation table was used, in which each subband weight was scaled by the standard deviation of that subband. Based on empirical evidence collected to date, we find that fixed-weight bit allocation may be more appropriate for SAR imagery.

The remainder of the paper is organized as follows. Section 2 contains a heuristic discussion of SAR image characteristics. We describe the compression algorithm in Section 3. Preliminary results, in terms of qualitative perceptual quality and image quality measures are presented in Section 4.

2 SAR Image Characteristics

SAR imagery is often characterized by a large dynamic range and a characteristic texture, typically referred to as “speckle.” As a result, SAR imagery typically has a large data entropy and is often much more difficult to compress than optical or computer-generated imagery. Specifically, electromagnetic scattering properties of man-made objects and natural terrain yield two characteristic features present in typical fine-resolution SAR imagery, *specular glints or flashes* and *speckle*. Specular returns appear as bright points or edges and typically arise from the radar returns from man-made objects, such as buildings and vehicles, and discrete clutter, such as tree trunks or rocks. Figure 1 shows a fine-resolution SAR image of part of a golf course. Present in the image are point-like specular returns from three trihedral reflectors along with edge-like returns from the roofs of two buildings.

Speckle is caused by diffuse scattering from surfaces that are rough compared to the wavelength of the radar [8]. Radar returns from natural terrain are often modeled as having a Rayleigh distribution with a parameter dependent on the mean terrain reflectivity. In Figure 1 one can see the edge between two different types of vegetated terrain.

Image analysts who work with fine resolution SAR imagery focus both on the image patterns caused by specular returns from man-made objects as well as the image texture caused by diffuse returns from natural terrain. In particular, the analyst may be required to perform object recognition, in which case the contextual information provided by the highly textured natural terrain may be just as important as the radar signature of a man-made object. Therefore, in order to preserve the analyst’s ability to interpret the imagery, it is important that both the edges and image texture are preserved. The approach we take is to separate the image into its specular and diffuse components and encode each separately using a perceptually-based bit allocation scheme.

2.1 Multiresolution Decomposition and Wavelet Shrinkage

A simple, nonparametric approach for extracting the edge information from imagery is to use wavelet shrinkage [6]. Donoho and Johnstone have shown that the wavelet transforms

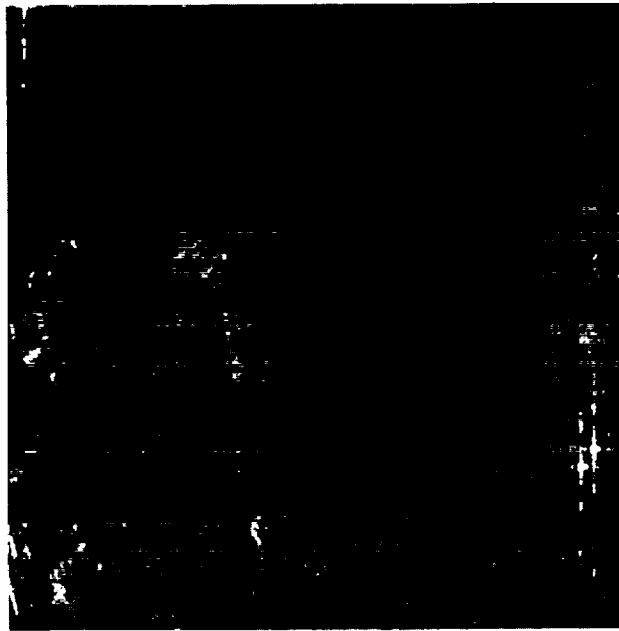


Figure 1: ADTS SAR image of a golf course. Specular returns can be seen from calibration trihedrals and buildings, while natural terrain yeilds diffuse returns (e.g., speckle).

of a broad class of functions, including piecewise-continuous functions, have a compact representation in the wavelet transform domain. On the other hand, an orthogonal discrete wavelet transform applied to white noise yields white noise having the same spectral density as before. Donoho and Johnstone propose a simple scheme for extracting smooth and piecewise-continuous signals from white noise: take the wavelet transform of the sampled noisy signal and apply a soft threshold to remove small wavelet transform coefficients that are likely to be noise.

In our context the speckle, or image texture in a SAR image, can be viewed as a nearly spatially-white but nonstationary noise process, while the edges, or specular returns, can be viewed as smooth or piecewise continuous functions. Figure 2 shows a multiresolution wavelet decomposition of the farm scene along with its decomposition into three components: local means, edges, and texture.

This decomposition is accomplished as follows. The four coefficient Daubechies filter [5] is used to perform a two-dimensional multiresolution wavelet decomposition of the SAR imagery. (Previous empirical evidence has shown that short-length wavelet filters are better than longer length filters for preserving points and edges in SAR imagery [18].) We use the decomposition specified by Mallat [12] to separate the image content according to spatial frequency and orientation. Throughout the remainder of the paper we will use the terminology of [12] and refer to subsets of the 2-D wavelet transform as “detail” images. The *local means* portion of our decomposition corresponds to the “coarse detail,” or lowest resolution detail image. The *edges* component consists of all wavelet coefficients exceeding the soft threshold or wavelet coefficient shrinkage operation [6]. Finally, the *texture* component is all of the remaining small coefficients.

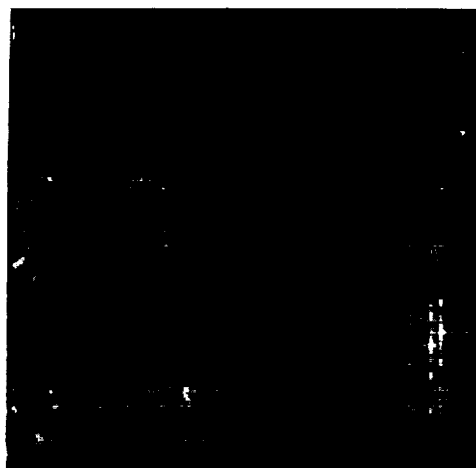
3 SAR Image Compression

We use the decomposition shown in Figure 2 as the basis for our compression algorithm. Figure 3 shows a schematic representation of the algorithm, which consists of four stages: a multiresolution wavelet transform (followed by gain normalization of the wavelet coefficients within each detail image), wavelet shrinkage to separate the image data into local means, edges, and textures, perceptually-based bit allocation based on a human visual system model (HVS), and a hybrid scalar/vector quantization operation.

After the 2-D wavelet decomposition has been performed, the coefficients of each detail image in the wavelet decomposition are gain normalized. Gain normalization allows the same vector quantizer to be used for multiple levels of the wavelet decomposition, and increases the efficiency of the vector quantizer. These normalization factors must be transmitted as side information.

Quantization bits are allocated to the wavelet coefficients according to human visual sensitivities to spatial frequency and spatial orientation, and according to whether the coefficients are edges, local means, or texture. The coefficients corresponding to the local means are allotted more bits than the texture coefficients. Moreover, a minimum rate is set for the edge coefficients so that when the overall data rate decreases, the edge coefficients are quantized and transmitted while the texture coefficients may not be transmitted at all. However, when the data rate is high, both edge and texture coefficients are allocated bits based upon

Multiresolution Wavelet Decomposition of a Magnitude-Detected SAR Image Into Three Sources:



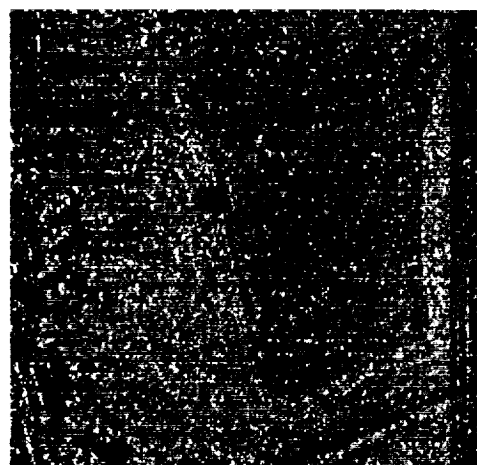
Original



Local Means



Edges



Texture

Figure 2: Decomposition of the ADTS image into local means, edges, texture components

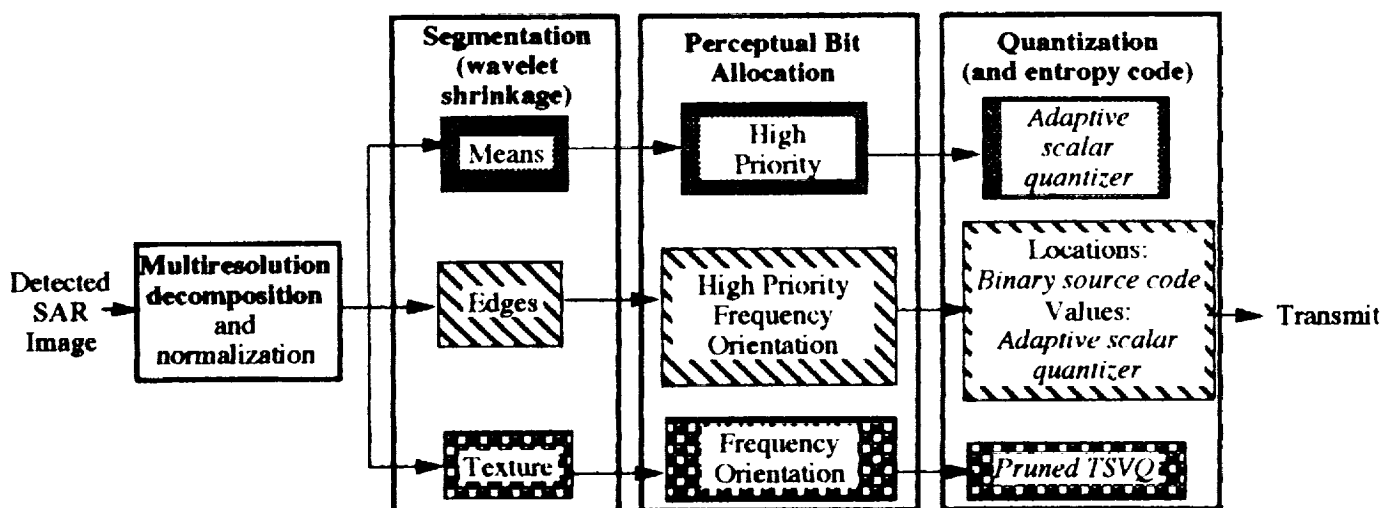


Figure 3: The perceptually-based compression algorithm consists of a wavelet multiresolution transform that is separated into local means, edges, and textures, followed by a hybrid scalar/vector quantizer with perceptually-based bit allocation.

perceptual sensitivity to spatial frequency and spatial orientation.

The bit allocation to spatial frequency and orientation differs from other HVS bit allocation methods in that it is completely independent of the statistics of the wavelet coefficients in each band. In other words, bits are allocated based solely on human visual system sensitivities rather than upon energy or mean-squared error considerations. The spatial frequency weights that are used for bit allocation are derived from equations developed for subband coding [14], which are based upon human contrast sensitivity data acquired by Campbell and Robson [2]. The equation used for bit allocation to each level of the multiresolution decomposition is given by:

$$B(k) = B_{tot} + \frac{1}{2} \log_2 \left[\left(W_{HVS}(k)^2 / A(k) \right) / \sigma_{HVS}^2 \right] \quad (1)$$

where $B(k)$ is the average number of bits allocated to detail image k , B_{tot} is the overall average bit rate, $W_{HVS}(k)$ is the human visual system weight obtained from the equation of Perkins and Lookabaugh [14], $A(k)$ is the relative area of detail image k , and σ_{HVS}^2 is a weighted geometric mean of the squared $W_{HVS}(k)$.

Vector quantizers (VQs) for 2×2 texture blocks were combined with adaptive scalar quantizers for edges and local means in a hybrid quantization scheme. The VQs we used were tree-structured variable-rate VQs [9] that were pruned using the optimal pruning algorithm of [4]. To maximize performance of the texture VQs, separate codebooks were created for the vertical, horizontal and diagonal texture components. As mentioned earlier, the edges and local means were quantized using high rate uniform scalar quantizers, while edge locations were coded using an error-resistant binary source coding technique [3]. The scalar quantizer step size was adapted in each detail image with dynamic range and wavelet shrinkage thresholds. Finally, the vector and scalar quantized coefficients were entropy coded.

4 An Example

The perceptual compression algorithm described above was applied to detected SAR imagery (remapped to 8 bpp) obtained from Lincoln Laboratory's Advanced Detection Technology Sensor (ADTS) System [10]. The resolution of this imagery is one foot in both the range and azimuth dimensions. Parameters for the HVS bit allocation and wavelet shrinkage threshold were determined by the viewing geometry, subjective evaluations, and available bit budget.

Figure 4 shows compressed versions of the farm scene at rates of 1, 1/2, and 1/4 bits per pixel (bpp). The visual quality of the SAR imagery compressed with the perceptual algorithm is excellent at moderate compression ratios (e.g. 8:1). As the compression ratio increases, the image quality degrades gracefully with minimal smearing of the edges and points. Even at very high compression ratios (e.g. 64:1), the images are recognizable. Also, there are no blockiness artifacts like those that are characteristic of the current version of the JPEG DCT algorithm [17] at rates below 1 bpp.

Finally, Figures 5 and 6 show plots of the measured impulse response (IPR) 3dB widths and image texture, as measured by coefficient of variation, for three different compression rates, 1, 1/2, and 1/4 bpp. Figure 5 contains a summary of several IPR measurements extracted from calibration trihedral signatures within the ATDS imagery. Both the mean

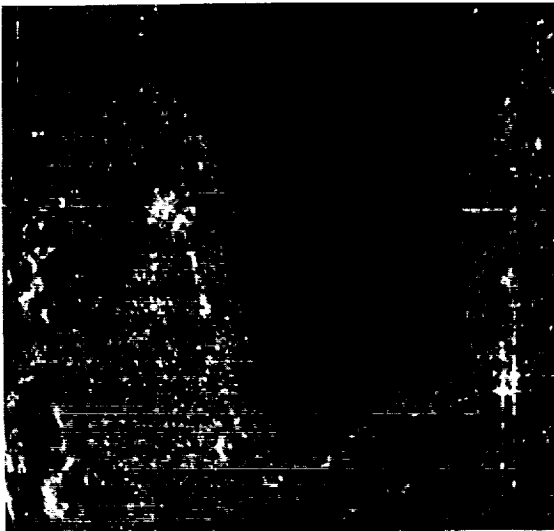
Perceptually-Based Multiresolution Compression of Magnitude-Detected SAR Imagery



Original (8 bpp)



Compressed to 1 bpp (8:1)



Compressed to 0.5 bpp (16:1)



Compressed to 0.25 bpp (32:1)

Figure 4: ADTS image compressed to 1, 1/2, and 1/4 bpp

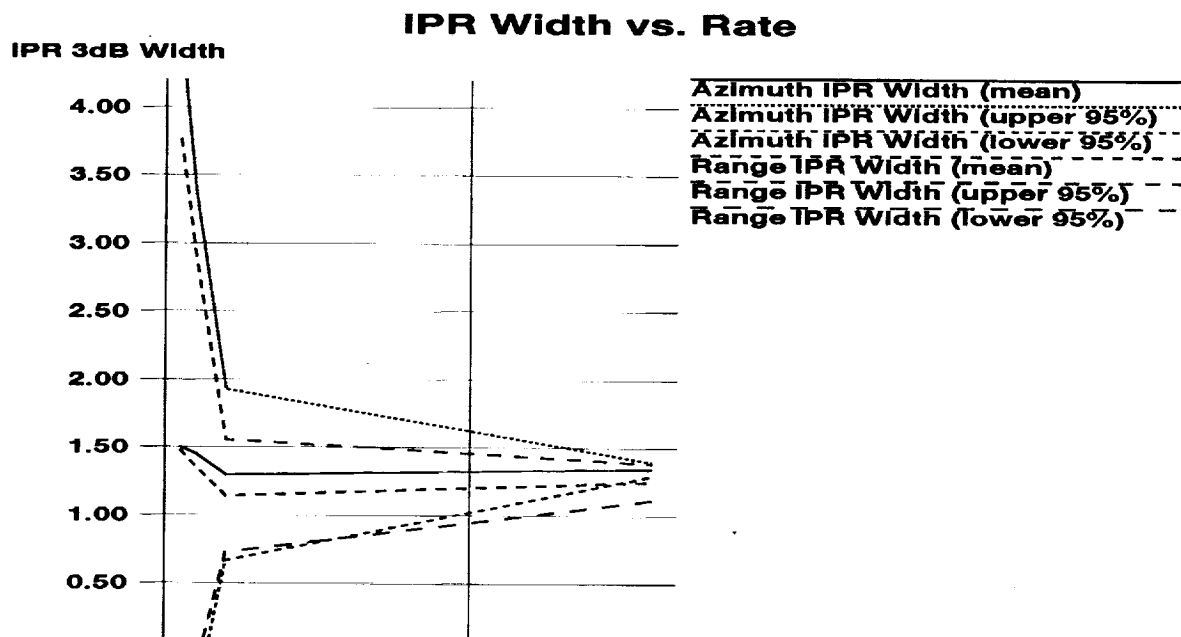


Figure 5: Impulse Response (IPR) 3dB Width Versus Data Rate.

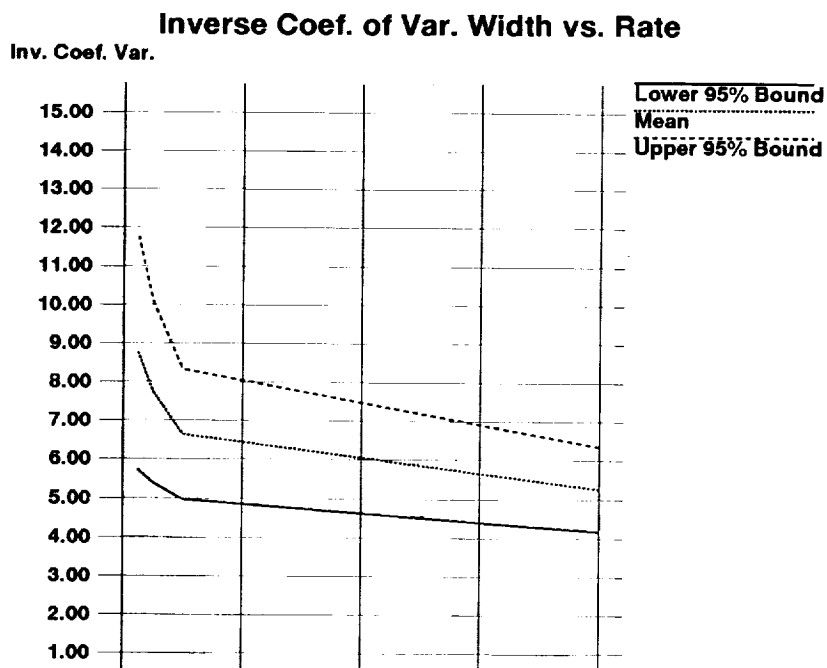


Figure 6: Inverse Coefficient of Variation Versus Data Rate.

IPR measurements in range and azimuth, along with 95% confidence bounds are plotted. What one can observe is that, on average, the IPRs only degrade from an original sampling rate of 1.3 samples per IPR to roughly 1.5 samples per IPR at a compression rate of 32:1 (i.e., 0.25 bpp). On the other hand, the variability of the IPR measurements increases dramatically as the data rate decreases.

Figure 6 shows a plot of the inverse coefficient of variation (mean divided by standard deviation) for a number of local measurements of terrain. Both the mean and upper and lower 95% confidence bounds are plotted for measurements taken over 144 different 15x15 pixel regions containing natural terrain. What we see is that as the data rate is decreased from 8 bpp (no compression) to 0.25 bpp, there is a loss of texture as measured by the increases in the inverse coefficient of variation. At 1 bpp there is a 26% increase in as compared to the original 8 bpp image, however, we observe no significant perceptual degradation. At 0.25 bpp, there is a 66% increase in the inverse coefficient of variation and noticeable smoothing of the image texture.

5 Summary

The perceptually-based multiresolution SAR compression algorithm presented here consists of a wavelet multiresolution decomposition followed by wavelet shrinkage, perceptually-based bit allocation, and hybrid scalar/vector quantization. An important feature that makes this particular approach appropriate for SAR imagery is the use of spatially-adaptive edge detection, via wavelet shrinkage techniques, to separate the image into three components: *local means*, *edges*, and *texture*. Each of these three components is then quantized separately using perceptual bit allocation mask. Based on preliminary results, we find that the algorithm provides excellent image quality at rates at or above 1 bpp and degrades gracefully below 1 bpp.

6 Acknowledgement

We would like to express our gratitude to Mr. John Henry of MIT Lincoln Laboratory for providing us with ADTS imagery.

References

- [1] Antonini, M., Barlaud, M., Mathieu, P. and Daubechies I., "Image Coding Using Wavelet Transform," *IEEE Trans. Image Proc.*, Vol. 1, pp. 205-220, 1992.
- [2] Campbell, F.W. and J.G. Robson, "Application of Fourier Analysis to the Visibility of Gratings," *J. Physiol.*, vol. 197, pp. 551-566, 1968.
- [3] Cheng, N.T. and N.G. Kingsbury, "The ERPC: An Efficient Error-Resilient Technique for Encoding Positional Information on Sparse Data," *IEEE Trans. Commun.*, Vol.40, No.1, pp.140-148, January 1992.

- [4] Chou, P. A., Lookabaugh, T., and Gray, R. M. "Optimal Pruning with Applications to Tree Structured Source Coding," *IEEE Trans. Info. Theory*, Vol. 35, pp. 299-315, 1989.
- [5] Daubechies, I. "Orthonormal Bases of Compactly Supported Wavelets," *Communication Pure Appl. Math*, Vol. 441, pp. 909-996, 1988.
- [6] Donoho, D. L. and Johnstone, I. M., "Ideal Spatial Adaption via Wavelet Shrinkage," Preprint, Technical Report No. 400, Department of Statistics, Stanford University, (available via anonymous ftp at stat.stanford.edu), April 1993.
- [7] Eddins, S.L., "Three-component image compression with frequency-weighted texture coding and edge modeling," *Proc. ICASSP*, pp. V369-V372, 1993.
- [8] Goodman, J.W., *Statistical Optics*, Wiley Interscience, 1985.
- [9] Gray, R. M. and Linde, J., "Vector Quantizers and Predictive Quantizers for Gauss-Markov Sources," *IEEE Trans. Comm.*, Vol. 30, pp. 381-389, 1982.
- [10] Henry, J. and P. Goetz, "35 GHz Images Created by the Advanced Detection Technology Sensor," *Tri-Service Radar Symposium*, pp. 157-161, 1988.
- [11] Lewis, A.S. and G. Knowles, "Image Compression Using the 2-D Wavelet Transform," *IEEE Trans. Image Processing*, Vol. 1, pp. 244-249, 1992.
- [12] Mallat, S. G., "A Theory for Multiresolution Signal Decomposition," *IEEE Trans. PAMI*, Vol. 11, pp. 674-693, 1989.
- [13] Marr, D., *Vision*, San Francicso, W.H. Freeman, 1982
- [14] Perkins, M.G. and T. Lookabaugh, "A Psychophysically Justified Bit Allocation Algorithm for Subband Image Coding Systems," *Proc. ICASSP*, pp. 1815-1818, 1989.
- [15] Safranek, R.J. and J.D. Johnston, "A Perceptually Tuned Subband Image Coder with Image Dependent Quantization and Post-Quantization Data Compression," *Proc. ICASSP*, pp. 1945-1948, 1989.
- [16] Shapiro, J.M., "Application of the Embedded Wavelet Hierarchical Image Coder to Very Low Bit Rate Image Coding," *Proc of IEEE ICASSP*, pp. 558-562, 1993.
- [17] Wallace, G., "The JPEG Still Picture Compression Standard," *Commun. ACM*, Vol. 34, pp. 31 - 45, 1991.
- [18] Werness, S. A., Wei, S. C., and Carpinella, R.J., "Experiments with Wavelets for Compression of SAR Data," *IEEE Trans. Geosci. Remote Sens.*, in press, 1993.

510 61
464
P 9

Compression of Multispectral Landsat Imagery Using the Embedded Zerotree Wavelet (EZW) Algorithm

Jerome M. Shapiro, Stephen A. Martucci, and Martin Czigler

The David Sarnoff Research Center

A Subsidiary of SRI International

Princeton, NJ 08543-5300.

Abstract

The Embedded Zerotree Wavelet (EZW) algorithm has proven to be an extremely efficient and flexible compression algorithm for low bit rate image coding [4]-[6]. The embedding algorithm attempts to order the bits in the bit stream in numerical importance and thus, a given code contains all lower rate encodings of the same algorithm. Thus, precise bit rate control is achievable and a target rate or distortion metric can be met exactly. Furthermore, the technique is fully image adaptive.

An algorithm for multispectral image compression which combines the spectral redundancy removal properties of the image-dependent Karhunen-Loeve Transform (KLT), with the efficiency, controllability and adaptivity of the Embedded Zerotree Wavelet algorithm is presented. Results are shown which illustrate the advantage of jointly encoding spectral components using the KLT and EZW.

1 Introduction

Multispectral image compression presents a set of new challenges in the area of image compression. In their raw form, multispectral images constitute a tremendous amount of data, and compression is essential for efficient data access, storage, and transmission of this class of imagery. Because there is also a large degree of interband correlation, there is potential for extremely high data compression without a large sacrifice in image quality, both subjectively and numerically.

In prior work described in [2], an image dependent Karhunen-Loeve Transform (KLT) was used to decorrelate a set of seven-band Landsat Thematic Mapper (TM) images prior to compression using a wavelet/subband coder. In the current work, the same image dependent KLT is used, but the compression engine that follows the KLT is replaced by a multiband implementation of the Embedded Zerotree Wavelet (EZW) algorithm. The EZW algorithm is a new compression algorithm that attempts to order the bits in the bit stream in numerical importance [4] - [6]. Because of the coarse to fine nature of the EZW algorithm, application to multiband images such as color or multispectral imagery involves simply including the additional wavelet coefficients for each band in the scanning used in EZW. This process is explained in more detail in Section 3.

2 Karhunen-Loeve Transform

There is typically a tremendous amount of interband correlation present in Landsat TM images since the sensors are co-located and the spectral weighting functions have some overlap. An effective way of exploiting this correlation is to compute the image-dependent KLT [2]. This involves performing an eigenvalue decomposition on the interband correlation matrix, and projecting the images, pixel-by-pixel, onto the orthonormal basis functions defined by the eigenvectors. The resulting principal component images each correspond to a different eigenvector. The amount of compression attainable depends on the eigenvalue spread, where a larger spread implies a higher coding gain. Once the interband correlation has been removed via the KLT, the resulting bands can be jointly encoded using the multiband EZW algorithm described in the next section.

Note that there is some overhead associated with the KLT that must be transmitted. In the results discussed below, the 7 means for each original band and the 49 elements of the eigenvector matrix are represented as 32-bit floating-point numbers for a fixed overhead of 1792 (56×32) bits. While this precision is probably unnecessary for large images, for example 512×512 , this overhead represents less than 0.007 bits per pixel. A larger drawback of the KLT approach is the computational burden in computing the KLT at the encoder. As discussed in [2], a fixed sub-optimal transformation, perhaps based on physical considerations, may be more practical at the cost of reduced coding gain. Alternatively, an intermediate compromise is to compute the KLT using data from the low frequency subbands of the wavelet transform for each original spectral component.

In addition to using the KLT for removal of spectral decomposition, Markas and Reif have also applied a histogram equalization technique to equalize the probability densities of the original bands [3]. Although this technique appears useful for visualization, the non-linearity effectively changes the gray scale units and amplifies the components with low spectral energy. As a result, joint bit allocation leads to unequal distortions distributed across the bands, causing the spectral components with the least energy to be encoded with the highest fidelity. Since EZW performs joint compression of all of the spectral components, unless the images are specifically compressed for visualization, histogram equalization would probably be inappropriate if uniform numerical distortion metrics are used.

3 Embedded Zerotree Wavelet Algorithm Description

3.1 Discrete Wavelet Transform

Each component is first transformed spatially using a discrete wavelet transform. The discrete wavelet transform used in this paper is identical to a hierarchical subband system, where the subbands are logarithmically spaced in frequency and represent an octave-band decomposition. This particular configuration has also been called a *QMF-pyramid* [1].

To begin the decomposition, the image is decomposed into four subbands by cascading horizontal and vertical two-channel critically sampled filterbanks. The filters used in the decomposition are scaled so that the squares of the filter coefficients sum to one. This normalization is important so that coefficients in all subbands can be compared to the same thresholds for the purpose of measuring numerical significance, since each coefficient is treated as a distinct, potentially important piece of data regardless of its scale. If orthogonal

wavelets are used, the resulting decomposition represents a unitary transformation. In practice, 9-tap symmetric QMF filters such as those in Adelson, *et. al.* [1] have been found to be effective. Note that for these QMF filters, the low-pass and high-pass filters in the filterbank are orthogonal, but these filters are only nearly orthogonal to their even-integer translates. However, for coding purposes, the discrete wavelet transform generated from these filters can be treated as unitary since the deviation from unitary is negligible compared to the quantization error.

After the first scale of the decomposition, to tile the entire image in each subband, each coefficient represents a spatial area corresponding to approximately a 2×2 area of the original picture. To tile the 2-D frequency domain, the low frequencies represent a bandwidth in each dimension approximately corresponding to $0 < |\omega| < \frac{\pi}{2}$, whereas the high frequencies represent the band from $\frac{\pi}{2} < |\omega| < \pi$. To obtain the next coarser scale of wavelet coefficients, the lowest frequency subband is further decomposed and critically sampled. The process continues until some final scale is reached. Note that at each scale, there are 3 subbands. The remaining lowest frequency subband is a representation of the information at all coarser scales. Note also that for each coarser scale, the coefficients represent a larger spatial area of the image but a narrower band of frequencies.

3.2 Successive-Approximation

To perform the embedded coding, successive-approximation quantization (SAQ) is applied. As will be seen, SAQ is related to bit-plane encoding of the magnitudes. Given an amplitude threshold T , a wavelet coefficient x is said to be *insignificant* with respect to T if $|x| < T$. The SAQ sequentially applies a sequence of thresholds $T_0, \dots, T_{N-1}$ to determine significance, where the thresholds are chosen so that $T_i = T_{i-1}/2$. The initial threshold T_0 is chosen so that $|x_j| < 2T_0$ for all transform coefficients x_j .

During the encoding (and decoding), two separate lists of coordinates of wavelet coefficients are maintained. At any point in the process, the *dominant* list contains the *coordinates* of those coefficients that have not yet been found to be significant in the same relative order as the initial scan. This scan is such that the subbands are ordered, and within each subband, the set of coefficients are ordered. The *subordinate* list contains the *magnitudes* of those coefficients that have been found to be significant. For each threshold, each list is scanned once.

3.3 The Dominant Pass: Zerotree Coding of Significance Maps

During a *dominant pass*, coefficients with coordinates on the dominant list, i.e. those that have not yet been found to be significant, are compared to the threshold T_i to determine their significance, and if significant, their sign is also recorded. A map indicating the result of a binary (significant or insignificant) or a ternary (positive significant, negative significant or insignificant) decision is called a *significance map*. This significance map for the dominant pass is encoded using zerotree coding as outlined below.

A parent-child relationship can be defined between wavelet coefficients at different scales corresponding to the same location. With the exception of the highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients at the next finer

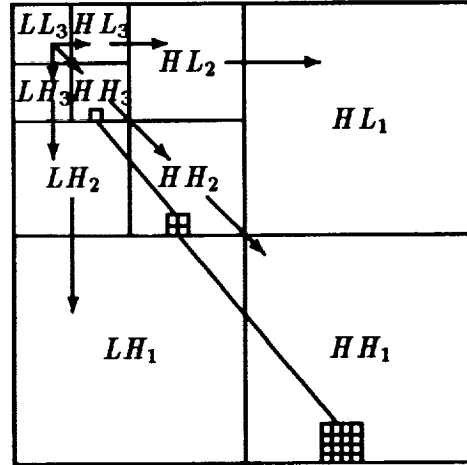


Figure 1: Parent-Child Dependencies of Subbands. Note that the arrow points from the subband of the parents to the subband of the children. The lowest frequency subband is the top left, and the highest frequency subband is at the bottom right. Also shown is a wavelet tree consisting of all of the descendants of a single coefficient in subband $HH3$. The coefficient in $HH3$ is a zerotree root if it is insignificant and *all* of its descendants are insignificant.

scale of similar orientation. The coefficient at the coarse scale will be called the *parent*, and all coefficients corresponding to the same spatial location at the next finer scale of similar orientation will be called *children*. The parent-child dependencies are shown in Fig. 1. With the exception of the lowest frequency subband, all parents have four children. For the lowest frequency subband, the parent-child relationship is defined such that each parent has three children, one in each subband at the same scale.

The scanning of the coefficients processed during a dominant pass is performed in such a way that no child is scanned before its parent. For an N -scale pyramid, the scan begins at the lowest frequency subband, denoted as LL_N , and scans subbands LH_N , HL_N , and HH_N , at which point it moves on to scale $N - 1$, etc. Note that each coefficient within a given subband is considered before the scan moves to the next subband.

Given a threshold level T_i to determine whether or not a coefficient is significant, a coefficient x is said to be an element of a *zerotree* if it is insignificant and all of its descendants are also insignificant. A coefficient is said to be a *zerotree root* for a threshold T_i if 1) the coefficient is insignificant, 2) the coefficient is not the descendant of a previously found zerotree root for T_i , i.e. it is not *predictably insignificant* from the discovery of a zerotree root at a coarser scale, and 3) *all* of its descendants are insignificant.

During the scanning of the coefficients during a dominant pass, each coefficient that is not predictably insignificant is encoded with a symbol from the four symbol alphabet: 1) zerotree root, 2) isolated zero, 3) positive significant, and 4) negative significant, where an isolated zero implies that the coefficient under consideration is insignificant but has a significant descendant. The string of symbols is then encoded using a multi-level adaptive arithmetic coder such as in Witten, *et. al* [7]. Each time a coefficient is encoded as significant,

(positive or negative), its magnitude is appended to the subordinate list. Also note that once a coefficient is determined to be significant, for the purpose of determining if one of its ancestors is a zerotree on future dominant passes, its value is treated as zero so as not to prevent a zerotree occurrence on future dominant passes.

3.4 The Subordinate Pass: Refinement of Significant Coefficients

A dominant pass is followed by a *subordinate pass* in which all coefficients on the subordinate list are scanned and the specifications of the magnitudes available to the decoder are refined to an additional bit of precision. More specifically, during a subordinate pass, the *width* of the effective quantizer step size, which defines an uncertainty interval for the true magnitude of the coefficient, is cut in half. For each magnitude on the subordinate list, this refinement can be encoded using a binary alphabet with a "1" symbol indicating that the true value falls in the upper half of the old uncertainty interval and a "0" symbol indicating the lower half. The string of symbols from this binary alphabet that is generated during a subordinate pass is then entropy coded. Note that prior to this refinement, the width of the uncertainty region is exactly equal to the current threshold. After the completion of a subordinate pass the magnitudes on the subordinate list are sorted in decreasing magnitude, to the extent that the decoder has the information to perform the same sort.

3.5 Embedded Coding

The process continues to alternate between dominant passes and subordinate passes where the threshold is halved before each dominant pass. (In principle one could divide by other factors than 2. This factor of 2 was chosen here because it has nice interpretations in terms of bit plane encoding and numerical precision in a familiar base 2, and good coding results were obtained).

In the decoding operation, each decoded symbol, both during a dominant and a subordinate pass, refines and reduces the width of the uncertainty interval in which the true value of the coefficient (or coefficients, in the case of a zerotree root) may occur. The reconstruction value used can be anywhere in that uncertainty interval. For minimum mean-square error distortion, one could use the centroid of the uncertainty region using some model for the PDF of the coefficients. However, a practical approach is to simply use the center of the uncertainty interval as the reconstruction value.

The encoding stops when some target stopping condition is met, such as when the bit budget is exhausted. The encoding can cease at any time and the resulting bit stream contains all lower rate encodings. Note that if the bit stream is truncated at an arbitrary point, there may be bits at the end of the code that do not decode to a valid symbol since a codeword has been truncated. In that case, these bits do not reduce the width of an uncertainty interval or any distortion function. In fact, it is very likely that the first L bits of the bit stream will produce exactly the same image as the first $L + 1$ bits which occurs if the additional bit is insufficient to complete the decoding of another symbol. Nevertheless, terminating the decoding of an embedded bit stream at a specific point in the bit stream produces exactly the same image would have resulted had that point been the initial target rate. This ability to cease encoding or decoding anywhere is extremely useful in systems

that are either rate-constrained or distortion-constrained. A side benefit of the technique is that an operational rate vs. distortion plot for the algorithm can be computed on-line.

Compression is achieved both by eliminating a large number of predictably insignificant coefficients from consideration through zerotree coding, and by adaptively arithmetic coding a string of symbols from a small alphabet. Note that the small size of the alphabet poses a tremendous advantage for an adaptive coder. Since all possible events usually occur with easily measurable frequency, an adaptation algorithm with a short memory can learn quickly and constantly track changing symbol probabilities. This adaptivity accounts for some of the effectiveness of the overall algorithm. Contrast this with the case of a large alphabet, as is the case in algorithms that don't use successive approximation. In that case, it takes many events before an extremely unlikely symbol occurs, and there are usually very many unlikely symbols. Furthermore, the probability estimates for rare events in a large alphabet are fairly unreliable because images are typically statistically non-stationary and local symbol probabilities change from region to region. Thus, the advantage of a small alphabet in an adaptive coder is that no coding capacity is wasted accounting for the possible occurrence of a large number of rare events.

3.6 Multiband EZW

Extension of the EZW algorithm to handle multispectral imagery is accomplished by simply including the wavelet transform of each principal component in the scan of the dominant pass. The scanning begins on the lowest frequency subband of the wavelet transform of the principal component corresponding to the largest eigenvalue. This entire component is scanned at a given threshold after which the scanning continues for each component in order of decreasing eigenvalue. Thus, a dominant pass for a given threshold involves scanning the transforms of all of the components at the same significance level. Although each component is scanned independently during a dominant pass, the magnitudes of significant coefficients are all placed on the *same* subordinate list. As a consequence, the refinement of significant coefficients on a subordinate pass makes no distinction as to which component a coefficient originated from. Although statistically the components corresponding to small eigenvalues contain little energy, if there are wavelet coefficients of these components that are large, bits will automatically be allocated to correctly represent their significance.

4 Experimental Results

The same Landsat 5 TM images of Kuwait that were used in [2] were again used in this new study. In addition, experiments were run using the Landsat images of Washington, D.C. All images were obtained from the USGS EROS Data Center (Sioux Falls, SD). As explained in [2], the Landsat TM data was produced by 7 sensors, where each sensor generates one band of imagery data. Bands 1 to 3 correspond to visible spectra, Band 4 to near IR spectra, Bands 5 and 7 to mid IR spectra, and Band 6 to thermal spectra. The instantaneous field of view (IFOV) for all sensors is about 30×30 m, except for Band 6, which has an IFOV of 120×120 m. All images are of size 512×512 pixels at 8 bits/pixel.

The sequence of steps for this new method of compressing multispectral data that were followed in this study are:

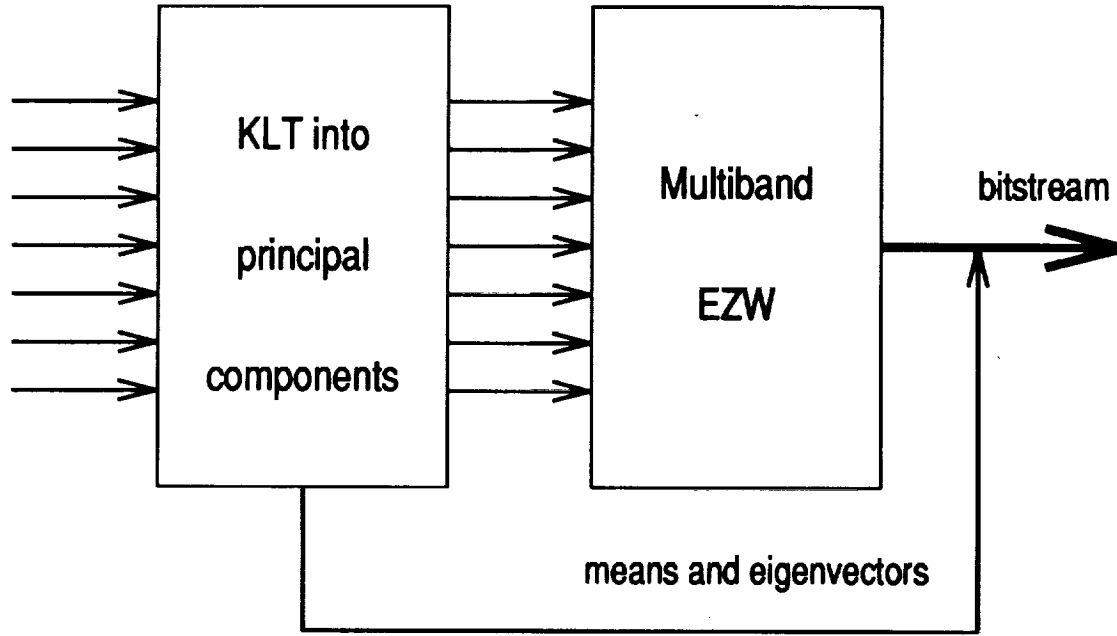


Figure 2: Block Diagram of KLT-EZW Encoder

1. Calculate then subtract the mean from each spectral band.
2. Calculate then apply KLT across all spectral bands to transform into principal components.
3. Compress principal components to target bit rate using the multispectral EZW algorithm.
4. Transmit means and eigenvectors as overhead.
5. Decompress bitstream using the multispectral EZW algorithm to recover the principal components.
6. Apply inverse KLT to transform principal components back into spectral bands.
7. Add mean to each band; reconstructed spectral bands result.

A block diagram of the encoder portion of the multispectral compression system is given in Fig. 2.

To evaluate the effectiveness of the new compression scheme, the mean square error between each original spectral band image and its reconstruction was calculated. These errors were then summed over all 7 bands. The totals are given in Table 1 for the Kuwait data under the heading *Principal Components* and subheading *new method* and in Table 2 for the Washington data under the heading *Principal Components*. The results reported in [2] are also included in Table 1 under the subheading *old method*. The bit rates shown in the table are the same as those reported in [2]. In that earlier study, the degree of compression was controlled by the specification of the quantizer bin sizes. Rate control was not used, and the bit rate of the encoded bitstream was just a consequence of the bin sizes. In the new

Table 1: Mean Square Error Results for Compression of Kuwait Images.

<i>bits/pixel</i>	<i>Original Bands</i>		<i>Principal Components</i>	
	<i>old method</i>	<i>new method</i>	<i>old method</i>	<i>new method</i>
2.51	40.02	31.63	N/A	14.13
1.55	N/A	52.04	25.11	20.24
1.06	73.82	62.45	N/A	29.16
0.73	N/A	83.71	47.96	38.28

Table 2: Mean Square Error Results for Compression of Washington Images.

<i>bits/pixel</i>	<i>Original Bands</i>	<i>Principal Components</i>
4.0	42.72	28.51
2.0	81.18	51.92
1.0	113.89	77.38
0.5	152.54	113.94

method, any desired bit rate can be met exactly; there is no need for explicit rate control. Thus, the mean square error results of the new method can be compared directly to those of the old method because the compression could be done to the same bit rates.

Experiments were also done to assess the performance of the multispectral EZW algorithm without first computing the principal components. The mean square errors of the resulting compressed images are given in the tables under the heading *Original Bands*.

As can be seen in the table, the new method gives significantly better performance than the old method, both when the principal components are not used and when they are. Even more significant is the improvement obtained by making use of the principal components. Thus, there are gains due to the multispectral EZW algorithm itself as well as gains due to transforming the imagery into its principal components.

5 Conclusion

Spectral decorrelation using an image dependent KLT followed by compression using the multiband EZW algorithm is an effective way to jointly encode the spectral bands of multispectral images. In contrast to the independent coding of the principal component images that was used in [2], the EZW algorithm jointly optimizes the bit allocation uniformly across all of the bands. Furthermore, the embedding and adaptivity features inherent in EZW allow precise rate control and eliminate the need to train the coder for a particular class of imagery.

References

- [1] E. H. Adelson, E. Simoncelli, and R. Hingorani, "Orthogonal pyramid transforms for image coding", *Proc. SPIE*, vol. 845, Cambridge, MA, pp. 50-58, Oct. 1987.
- [2] B. R. Epstein, R. Hingorani, J. M. Shapiro, and M. Czigler, "Multispectral KLT-Wavelet Data Compression for Landsat Thematic Mapper Images", *Proc. Data Compression Conf.*, Snowbird, Utah, IEEE Computer Society Press, 1992.
- [3] T. Markas and J. Reif, "Multispectral Image Compression Algorithm", *Proc. Data Compression Conf.*, Snowbird, Utah, IEEE Computer Society Press, 1993.
- [4] J. M. Shapiro, "An Embedded Wavelet Hierarchical Image Coder", *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, San Francisco, CA, March 1992.
- [5] J. M. Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients", *Proc. Data Compression Conf.*, Snowbird, Utah, IEEE Computer Society Press, 1993.
- [6] J. M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients", to appear *IEEE Trans. Signal Processing, Special Issue on Wavelets and Signal Processing*, scheduled for Dec. 1993.
- [7] I. H. Witten, R. Neal, and J. G. Cleary, "Arithmetic coding for data compression", *Comm. ACM*, vol. 30, pp. 520-540, June 1987.

Vector Quantizer Designs for Joint Compression and Terrain Categorization of Multispectral Imagery¹

John D. Gorman

Environmental Research Institute of Michigan, Box 134001, Ann Arbor, MI, 48113-4001

Voice: (313) 994-1200, FAX: (313) 994-5704, Email: gorman@erim.org

and

Daniel F. Lyons

The Mitre Corporation, M/S W622, 7525 Colshire Drive, McLean, VA 22102-3481

Voice: (703) 883-1916, FAX: (703) 883-6708, Email: dflyons@oceanus.mitre.org

ABSTRACT

We evaluate two vector quantizer designs for compression of multispectral imagery and their impact on terrain categorization performance. The mean-squared error (MSE) and classification performance of the two quantizers are compared, and it is shown that a simple two-stage design minimizing MSE subject to a constraint on classification performance has a significantly better classification performance than a standard MSE-based tree-structured vector quantizer followed by maximum-likelihood classification. This improvement in classification performance is obtained with minimal loss in MSE performance. Our results show that it is advantageous to tailor compression algorithm designs to the required data exploitation tasks. Applications of joint compression/classification include compression for the archival or transmission of Landsat imagery that is later used for land utility surveys and/or radiometric analysis.

1 Introduction

The vast majority of vector quantizer (VQ) design algorithms presume the use of mean-squared error (MSE) as a metric. The shortcomings of MSE on perceptual quality in image coding are well known. In this paper, we show that MSE-based quantization severely degrades the performance of M -ary classification algorithms following compression and decompression. Appropriate design criteria for the joint compression and classification problem should include some combination of MSE and Bayes risk. In the context of multispectral imagery, MSE is a reasonable criterion for quantizers that are designed to preserve the root mean-squared (RMS) radiometric accuracy of the imagery. Bayes risk, on the other hand, is appropriate for designs that optimize terrain categorization performance, since it directly relates to classification performance.

We explore two vector quantizer designs, an *independent* design and a *joint* design. The independent design uses a standard MSE-based tree-structured vector quantizer (TSVQ) followed by a maximum-likelihood classifier that optimizes probability of correct classification

¹This work supported in part under internal ERIM funding.

[5]. The joint design, on the other hand, optimizes MSE performance subject to a constraint on classification performance. For this latter design, a two-stage quantizer is used [6,7]. The first quantization stage is a tree-structured classifier (TSC) [1,2] that essentially performs a coarse quantization of the multispectral pixel feature space. This coarse quantizer is then refined using a second quantizer that is designed using a MSE criterion. An alternative to the joint compression/classification problem has recently been proposed by Cosman et. al. [3].

We present results on the MSE and terrain categorization performance of these two quantizer designs at various information compression rates for Landsat-4 Thematic Mapper data collected over Ann Arbor, Michigan are presented. Empirical results indicate that the joint design provides superior classification performance with minimal MSE degradation.

2 Results

We demonstrate that for MSE-based TSVQ codebook designs having large or even moderate compression ratios of 8:1 or better, classification performance on compressed imagery is severely degraded relative to the performance of the classical maximum-likelihood classifier operating on uncompressed imagery. This performance degradation is due to the fact that at high compression ratios (that is, low code rates), there is a tendency for classes having large component variances to mask other classes that have smaller variances—even when the classes are well separated. This is because the MSE criterion protects against large errors regardless of the resulting classification performance.

Figure 1 shows a scatter plot from two bands of Landsat-4 multispectral data for a simple four-class problem; band 5 radiances are plotted against the corresponding band 3 radiances for four terrain categories: clouds, soil, water and wetlands. Two different algorithms were used to partition the scatter plot into four regions. The partition selected by an MSE-based TSVQ is shown in solid lines while the partition selected by a tree-structured classifier is shown in dashed lines. Also shown in Figure 1 are the corresponding *codewords*: each data point falling into a given partition element is represented by the codeword for that partition element.

In Figure 1, the large-variance class (clouds) is “over coded.” In the MSE-based partition, the soil and wetland classes are not distinguished since they fall into a single partition element. In this case, compression of the data with the TSVQ would result in a loss of classification performance. Nonetheless, the four classes are well separated and a classifier partition can be designed to separate all four classes. Indeed, the classifier partition allows each of the four terrain categories to be distinguished.

The independent and joint compression/terrain categorization designs were applied to the six reflective bands from a 185×185 km² Landsat-4 frame collected over the southeast

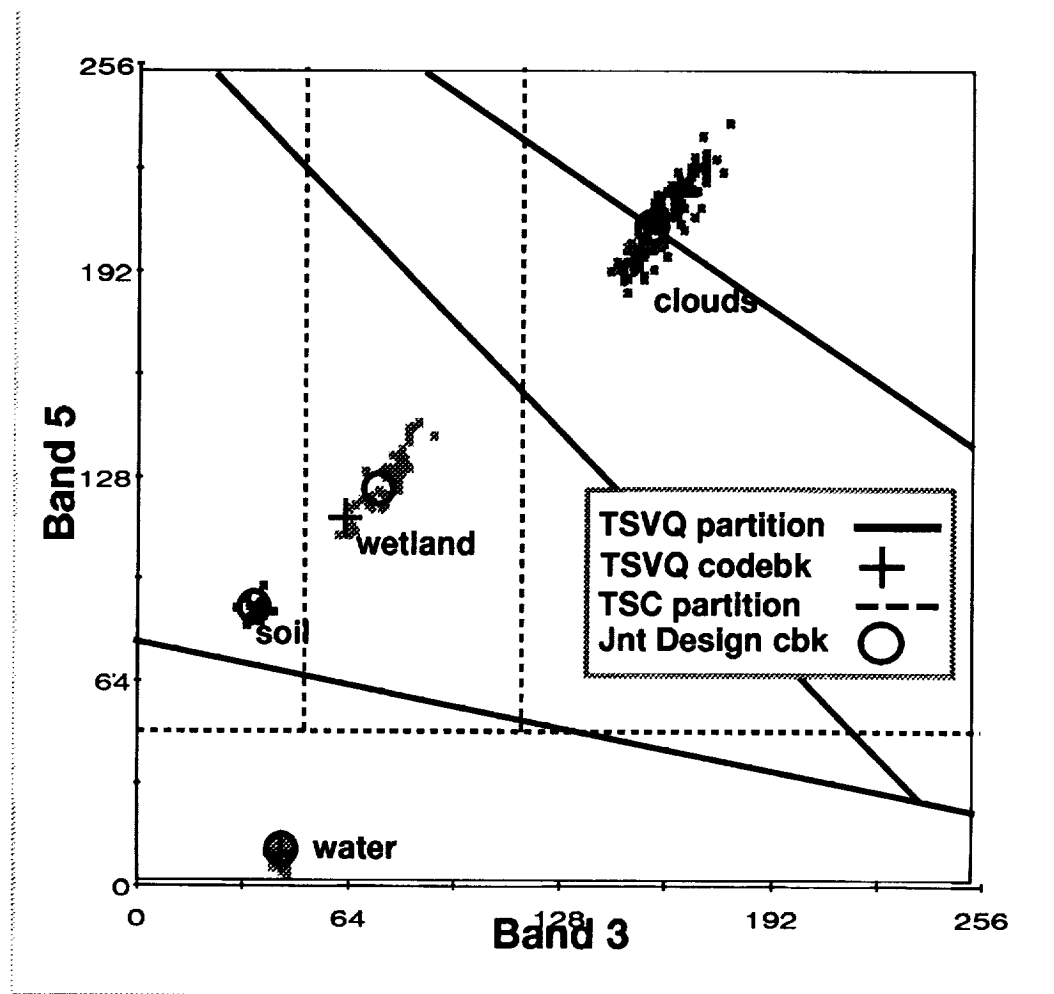


Figure 1. Two feature-space partitions for the four-class terrain categorization example.

Michigan area. A total of 10 general terrain classes: urban, agricultural, bare soil, range, deciduous, conifer, water, barren and cloud covered were located and identified by an experienced image interpreter. Figure 2 shows a quantitative performance comparison between the independent and joint design approaches. Specifically, Figure 2 shows both the MSE and classification error rate as a function of the code rate. The classification error rate is computed with respect to the terrain categorization performance on the original data. The various curves in Figure 2 show the performance of four VQ designs: the independent design, and joint designs in which the first stage (i.e., the classifier) is allocated 5, 6, and 8 bits. The original data rate is 48 bits per pixel (bpp) (i.e., six bands at 8 bits/band/pixel). The plots show that a substantial rate decrease can be achieved while still retaining the same classification error rate. In particular, at a 4:1 compression rate, or 12 bpp, the joint scheme has a 4% RMS radiometric error and a 2% classification error. This should be compared to the independent scheme which has a slightly lower RMS radiometric error of 0.5%, but a significantly larger classification error of 25%.

Finally, Figure 3 shows the output of the terrain categorization step after compression at a 12:1 compression ratio (i.e., a data rate of 4 bits/multispectral pixel). Figure 3a shows the original classification output. Figure 3b shows the output of the independent compression/classification design (i.e., the supervised maximum-likelihood classifier operating on data that has been compressed 12:1 with a MSE-based tree-structured vector quantizer). Figure 3c shows the output of the joint compression/classification design. In the independent design, the water category is classified as a non-category, while many of the other classes are missing completely. On the other hand, in the joint design much of the original spatial structure in the classification map is preserved and the classification errors are spatially localized. In fact, when we examined the difference between the joint design output in Figure 3c and the original classifier output in Figure 3a, we found that approximately 93% of the classification errors occurred over regions that were 3×3 pixels across or smaller.

3 Conclusions

We compared two quantizer designs for the problem of joint compression/terrain categorization of multispectral imagery. The first quantizer design was an *independent* design, consisting of a mean-squared error (MSE) based quantizer design followed by a maximum-likelihood classifier. The second design was a *joint* design that employed a two-stage quantizer. The first stage consisted of a tree-structured classifier that performed a coarse quantization of the image data. This coarse quantization was then refined using a standard MSE-based tree-structured quantizer. One can view this two-stage process as one particular approach to minimizing MSE subject to a constraint on allowable classification error.

We showed that the joint design achieved a significant improvement in classification per-

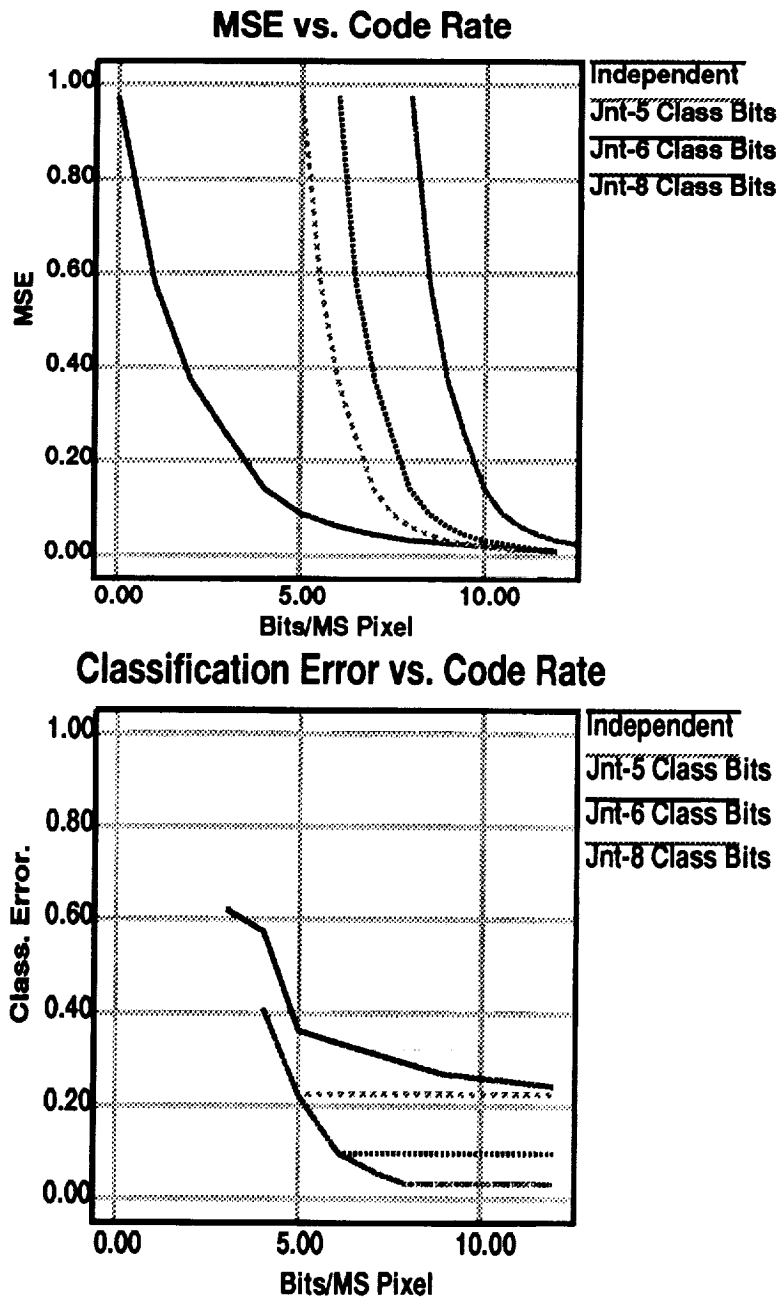


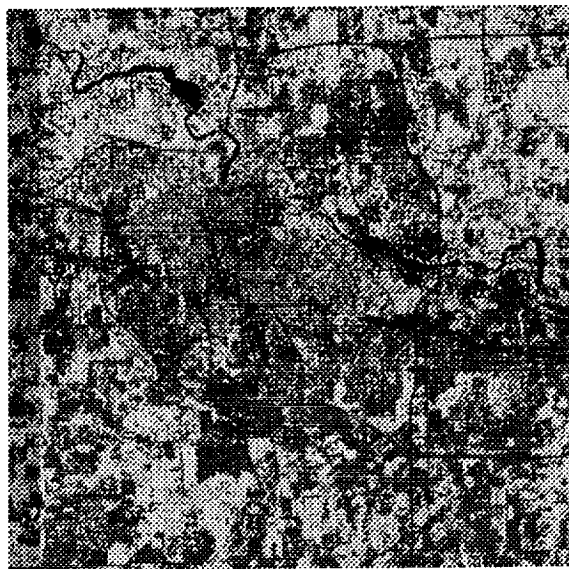
Figure 2. MSE and Classification performance of the two compression/ classification schemes



(a) Original Classification



(b) ML Classifier, 12:1 Compression



(c) TSC, 12:1 Compression

- water
- urban
- range
- agricult.
- bare soil
- deciduous
- conifer
- quarry
- barren
- golf course
- non category

Figure 3. Joint compression/terrain categorization examples: (a) Original terrain categorization, (b) independent compression/classification designs, (c) joint classification/compression design.

formance with only a minor degradation in MSE performance. This suggests that significant increases in data exploitation utility can be achieved by modifying compression algorithm design criteria to include metrics appropriate to the required exploitation tasks.

References

- [1] L. Breiman, Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*, Wadsworth Inc., Pacific Grove, Cal. 1984.
- [2] W. Buntine, *Introduction to IND and Recursive Partitioning*, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffet Field, CA, June 23, 1992.
- [3] Cosman, P. C., Oehler, K. L., Riskin, E. A. and Gray, R. M., "Using Vector Quantization for Image Processing," *Proc. IEEE*, Vol. 81, No. 9, pp. 1326-1343, September, 1993.
- [4] Fukunaga, K., *Pattern Recognition*, 2nd ed., 1991.
- [5] Gersho, A. and Gray, R. M., *Vector Quantization*, Kluwer Academic, Norwell, MA, 1992.
- [6] Gorman, J. D. and Lyons, D. F., *Proc. 26th Conf. on Info. Sci. and Sys*, Princeton, NJ, 1991.
- [7] Gorman, J. D., Subotic, N. S. and Robinson, S. R., "Impact of Compression on Terrain Categorization Performance of Multispectral Data," *Proc. 1992 Int'l Spectral Sensing for Space Research*, Maui, Hawaii, November 1992.

Wavelet Compression Techniques for Hyperspectral Data

Bruce Evans, Brian Ringer and Mathew Yeates

Data Technologies Division

TRW Systems Integration Group

One Space Park, Bldg. R2 / 2170

Redondo Beach, CA 90278

cuber@spf.trw.com

512-61

466

P-11

Hyperspectral sensors are electro-optic sensors which typically operate in visible and near infrared bands. Their characteristic property is the ability to resolve a relatively large number (i.e., tens to hundreds) of contiguous spectral bands to produce a detailed profile of the electromagnetic spectrum. In contrast, multispectral sensors measure relatively few non-contiguous spectral bands. Like multispectral sensors, hyperspectral sensors are often also imaging sensors, measuring spectra over an array of spatial resolution cells. The data produced may thus be viewed as a three dimensional array of samples in which two dimensions correspond to spatial position and the third to wavelength.

Because they multiply the already large storage/transmission bandwidth requirements of conventional digital images, hyperspectral sensors generate formidable torrents of data. Their fine spectral resolution typically results in high redundancy in the spectral dimension, so that hyperspectral data sets are excellent candidates for compression. Although there have been a number of studies of compression algorithms for multispectral data [1,2,3,4], we are not aware of any published results for hyperspectral data.

In this paper we compare three algorithms for hyperspectral data compression. They were selected as representatives of three major approaches for extending conventional lossy image compression techniques to hyperspectral data. The simplest approach treats the data as an ensemble of images and compresses each image independently, ignoring the correlation between spectral bands. The second approach transforms the data to decorrelate the spectral bands, and then compresses the transformed data as a set of independent images. The third approach directly generalizes two-dimensional transform coding by applying a three-dimensional transform as part of the usual transform-quantize-entropy code procedure. The algorithms studied all use the discrete wavelet transform. In the first two cases, a wavelet transform coder (using the algorithm described in [5]) was used for the two-dimensional compression. The third case used a three dimensional extension of this same algorithm.

These algorithms were tested on several data sets obtained from the TRW imaging spectrometer (TRWIS). This sensor provides measurements from 90 uniform width spectral bands which cover a wavelength range from approximately 400 nm to 800 nm, and is mounted in a helicopter or small plane. Spectra are obtained simultaneously from a linear array of 256 spatial resolution cells. Platform motion is utilized to scan this array, thus obtaining spatial samples in a second spatial dimension. A typical TRWIS data set consists of a 90x256x450 array of one byte samples.

Although signal to noise ratio (SNR) and related mean square distortion metrics are convenient and widely used, their relevance to practical utility or perceptual quality is uncertain. This is of particular concern with respect to hyperspectral data, since the art of interpreting and utilizing this data is still developing. To supplement SNR measurements for the different algorithms, we also applied example pixel classification and image segmentation algorithms to the reconstructed data sets in order to assess the impact of compression losses on automatic data exploitation. These applications include pixel classification using a k-means algorithm and region based spectral image segmentation.

Our results showed substantial differences in the performance of the three algorithms. The spectral decorrelation algorithm produces the best results, but also requires the most computational effort. The three dimensional wavelet algorithm's performance came in second, but well ahead of the band independent algorithm. These results clearly demonstrate the importance of exploiting the spectral redundancy. Spectral decorrelation performs best because the transform is optimally matched to the data, whereas the wavelet transform is suboptimal but computationally

more efficient. Interestingly, individual spectral bands displayed as images often look better in the reconstructed data than the original image, particularly for the spectral decorrelation algorithm. This is because the compression process in effect filters out sensor noise from the original signal.

Acknowledgments. This research was supported by TRW's Ballistic Missile Division through the coordination of Lou Cassel and Joan Lurie. Hyperspectral data and computational facilities were provided by TRW's Remote Sensing Center, directed by Stokes Fishburne.

Band-independent Wavelet Compression. This algorithm was primarily studied as a reference point for measuring the gains due to inter band processing. One advantage is that individual bands can be reconstructed without having to decompress the entire data cube. This is useful if one knows in advance that only a few spectral bands will be reconstructed from the compressed data, but not specifically which bands.

The performance of this algorithm of course depends entirely on the algorithm used to compress the individual bands. We selected the wavelet transform coding algorithm in [5] because our previous studies had shown its performance to be superior to DCT and DPCM algorithms and comparable to other wavelet algorithms. This algorithm first computes the discrete wavelet transform of the image using the Mallat [9] recursion and a Daubechies 4-tap wavelet kernel [8]. The transform is then partitioned into a collection of rectangular blocks, and quantizer bit rates are optimally assigned to each block using the algorithm described in [6,7]. The quantized coefficients are Huffman coded, and the side data consisting of the bit rate allocations for each block is losslessly compressed using the UNIX "compress" utility.

Three Dimensional Wavelet Transform Compression. This algorithm is a straightforward generalization of the two dimensional algorithm described above. All the components of the two dimensional algorithm have obvious three dimensional analogs; the major difficulty is the more complex bookkeeping required to manage three dimensional data. Our implementation emphasized simplicity and flexibility over efficiency, relying instead on a powerful workstation (a Sun SPARC 10), plenty of memory, and patience. However, hyperspectral data sets are generally large (around 36MB in our examples) and despite the algorithm's moderate complexity, processing can be time consuming. We expect that optimizing the implementation, particularly by improving memory management, would speed computation significantly, even on fast machines with large memories.

The three dimensional wavelet transform is constructed as a separable extension of the two dimensional transform, much as the two dimensional transform can be constructed by applying one dimensional wavelet filter banks over each dimension. Each stage of the separable three dimensional transform applies one dimensional filter banks successively across the two spatial dimensions and the spectral dimension. This decomposes the data into seven highpass channels and one lowpass channel. The seven highpass channels contain oriented edge information (in the two spatial directions, the spectral direction and the four diagonal combinations of these directions). Each channel contains one eighth of the original number of samples. Applying this operation recursively to the lowpass channel produces a series of nested octant decompositions.

We quantize the transform coefficients by partitioning each channel at each scale into three-dimensional sub-blocks. Within a sub-block, coefficients are quantized with the same number of bits per sample. Because large magnitude high pass coefficients tend to be sparsely distributed, many blocks can be quantized at low bit rates while introducing little distortion as a result. The actual bit allocation is determined using the algorithm described in [6,7]. This algorithm assumes that the mean square quantizer distortion is an exponential function of the bit rate times the sample variance of the data. It produces a bit allocation which minimizes the mean square quantization error subject to a constraint on the maximum average bit rate.

As in the two dimensional algorithm the quantized coefficients are Huffman coded. One difference is that three dimensional case uses a Lloyd-Max quantizer which is optimized for-each data set, and Huffman codes are determined based on the actual sample distributions for each bit rate. The two dimensional algorithm uses a fixed uniform quantizer and fixed Huffman codes (both optimized for Laplacian statistics). For large hyperspectral data sets, the additional side data needed to transmit the quantizer coefficients and Huffman code tables is relatively insignificant.

The side data also contains the quantizer bit rate allocations, and is compressed using the UNIX "compress" utility.

Band Decorrelation Wavelet Compression. The compression algorithm consists of the following steps. First we organize the data as a collection of spectral vectors $D = \{\mathbf{d}_{k,l}\}$, where a spectral vector $\mathbf{d}_{k,l}$ consists of all spectral samples corresponding to spatial resolution cell (k,l) . The spectral vectors lie in an n -dimensional Euclidean space, where n is the number of spectral bands. To each vector in D we then apply an affine transformation $T: \mathbf{d}_{k,l} \mapsto T(\mathbf{d}_{k,l}) \equiv \mathbf{c}_{k,l}$, where $\mathbf{c}_{k,l}$ has dimension $m < n$, to produce the transformed data set

$C = \{\mathbf{c}_{k,l}\}$. This data set is then compressed on a band by band basis using two dimensional wavelet coding as described above, with one key difference. The band independent algorithm compresses each spectral band to the same bit rate, but the band decorrelation algorithm varies the bit rate from plane to plane (subject to an upper bound constraint on the average bit rate). This is done because the transformation T concentrates most of the energy in a few spectral bands, so that allocating higher bit rates to these bands (and correspondingly lower rates to lower energy bands) significantly reduces distortion. The bit allocation is determined by the optimal algorithm described in [6,7]. This algorithm minimizes distortion assuming that the band compression algorithm has an exponential bit rate vs. mean square distortion curve with amplitude proportional to the sum squared in-band energy, and assigns bit rates to bands in proportion to their log-sum-square energy.

To reconstruct the data, C is reconstructed from the wavelet encoding for each band, and then the pseudo-inverse transformation $T^T: \mathbf{c}_{k,l} \mapsto T^T(\mathbf{c}_{k,l}) \equiv \hat{\mathbf{d}}_{k,l}$ is applied to reconstruct the original data. Note that distortion is introduced both from the lossy wavelet coding and because the transform T generally has no true inverse. However, the pseudo-inverse transform spreads reconstruction errors in C over many spectral bands, making them much less perceptible. Furthermore, the decorrelation transform is structured to minimize the loss of information due to its singularity.

Although we use the well known discrete Karhunen-Loeve transformation (or principle components analysis) for spectral decorrelation, we feel it is worthwhile to outline a derivation of this transform from a physical and geometric approach that may be less familiar than the statistical approach. This approach shows that the transform is optimal in a sense that does not depend on statistical assumptions that may be hard to justify in practice. It also provides insight into the effectiveness of this transform for compression.

We assume that the spectra in any given data set are primarily linear combinations of spectra corresponding to the various materials constituting the scene. Generally, the number different materials is much less than the number of spectral bands. We therefore expect most of the spectral vectors to lie in, or close to, a linear subspace whose dimension is much lower than the dimension of the spectral vectors. If we could find the basis vectors for this space, then we could produce a lower dimensional approximation by projecting the original spectral vectors onto this space.

Stated more precisely, given a collection $D = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_p\}$ of data vectors in a n -dimensional linear space L , we wish to find a set of m orthonormal n -vectors (with $m < n$), spanning a subspace S of L , such that the sum of the squared distances between each data vector in L and its orthogonal projection onto S is minimized. If we define the sample autocovariance matrix $\mathbf{R} = \sum_{k=1}^p \mathbf{d}_k \mathbf{d}_k^T$ it can be shown that the required basis vectors are the unit eigenvectors $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m\}$ corresponding to the m largest eigenvalues of $\mathbf{R}$. Note that the coordinates in S of the projection any $\mathbf{d}$ in L onto S are simply its inner products with the basis vectors, $(\mathbf{e}_1^T \mathbf{d}, \mathbf{e}_2^T \mathbf{d}, \dots, \mathbf{e}_m^T \mathbf{d})$. Furthermore, any vector $\mathbf{c}$ in S with coordinates $(c_1, \dots, c_m)$ can be

represented in L as a linear combination of the basis vectors $\mathbf{c} = \sum_{k=1}^m c_k \mathbf{e}_k$. We thus have the transform $T:L \rightarrow S$ represented by the matrix $\mathbf{T}$ whose rows are the (transposed) basis vectors of S , i.e. $T(\mathbf{d}) = \mathbf{T}\mathbf{d}$. Furthermore, this transformation has the pseudo-inverse $T^\dagger:S \rightarrow L$ with $T^\dagger(\mathbf{c}) = \mathbf{T}^T \mathbf{c}$.

Note that in our algorithm, $\mathbf{T}$ is determined specifically for each data set, based on the sample autocovariance $\mathbf{R}$. Some spectral decorrelation algorithms, such as [1] use a fixed $\mathbf{T}$ derived from statistical model that is independent of the actual data. Although this saves computation, it sacrifices the optimality of the transform. Computing $\mathbf{T}$ might appear burdensome, but for hyperspectral data the effort required to *apply* $\mathbf{T}$ is typically many times the effort of the eigensystem solution needed to *find* $\mathbf{T}$. A more serious objection may be that $\mathbf{T}$ must be sent as side data in order to decompress the data.

As a corollary of the construction of $\mathbf{T}$, it turns out that the eigenvalues of $\mathbf{R}$ corresponding to basis vectors in S equal the sum of squares of the coefficients in the corresponding "spectral" band of the transformed data set C . The fact is quite useful because these sum squared band energies are the statistics required to allocate average quantizer bit rates to each band. This means that these bit allocations be determined before the spectral decorrelation transform is actually applied. As a result, rows corresponding to zero or near zero bit rates can simply be dropped from $\mathbf{T}$, significantly reducing the number of operations required to compute the transform.

Experimental results. We present results for two data sets produced by the TRWIS sensor. These data sets each contain 90 uniformly spaced and contiguous spectral bands, spanning a wavelength range of 400 to 800 nm. Within each spectral band, there are 450 raster lines with 236 samples per line, with eight bit deep samples. They have been calibrated to compensate for variations in illumination intensity with bandwidth, so that the samples actually represent estimated percent reflectance. Consequently, one expects sample values between zero and 100, but because the calibration is with respect to a diffuse white reference reflector, specular reflections can produce values above 100. Figures 1 and 2 show images from one spectral band in each of these data sets. The first data set ("houses") shows a residential area with houses and vegetation. The second data set ("tents") is an aerial view of tents and military vehicles on a sandy background.

Figure 3 shows plots of peak-signal to noise ratio (PSNR) as a function of compression ratio for each data set and each algorithm. We define PSNR as the square of maximum sample value in the original data set divided by the mean squared error between the original and reconstructed images. The vertical scale in the figure shows PSNR in decibel units. The horizontal scale shows the ratio of the original file size to the compressed file size. For every algorithm, the "tents" PSNR is higher than the corresponding PSNR for the "houses", which reflects the greater compressibility of this image. Other than this uniform vertical shift, the results for the two data sets are quite similar. Substantial differences between the algorithms are evident. The PSNR for the 3-D wavelet transform is two to three dB higher than the band independent algorithm, and in turn the spectral decorrelation PSNR exceeds the 3-D wavelet transform by about four dB.

Comparisons of spectral band images clearly reflect the differences in the rate-distortion curves. Figure 4 shows images of the same spectral band from original and compressed/reconstructed versions of the "houses" data set. The band independent algorithm was used for the top row of images, the 3-D wavelet algorithm for the middle row, and the band decorrelation algorithm for the bottom row. Within each row, the leftmost image is the original data, and the three remaining images correspond to increasing compression ratios from left to right. The spectral decorrelation images are clearly much less distorted than the others. When viewed on a high quality display, distinguishing the reconstructed spectral decorrelation image from the original requires close observation, even at the highest compression ratio. In the case of the 3-D wavelet transform, many of the fine, high contrast details are preserved fairly well, but

there is a noticeable loss of texture and detail in low contrast regions. At the highest compression, these losses are quite obvious. The quality of the best band independent reconstruction appears to be about equivalent to the worst 3-D reconstruction. At the highest compression, all detail is lost, although high contrast edges are fairly well preserved.

Examining the spectral decorrelation images reveals some interesting effects. Although distortion is almost imperceptible, at the highest compression ratio there are a few regions in which there are systematic shifts in the gray levels at which certain features in original data are reproduced. (E.g., the small, crescent shaped dark area immediately below the house at the center left of the image and a curved, dark area contained within a bright, semi-elliptical area at the center of the right edge). These areas apparently contain materials whose reflection spectra are outside the subspace spanned by the spectral decorrelation basis. Since the basis is selected to optimize a mean squared criterion, small or infrequently occurring spectra tend to be poorly represented. As a consequence, in applications where one wishes to detect spectral shapes that are sparsely represented in the original image (such as finding a few camouflaged tents in a forest), spectral decorrelation may perform poorly despite producing excellent mean square error based figures of merit, such as PSNR. In contrast, the band independent and 3-D wavelet algorithms appear to be free of such systematic gray level shifts.

This illustrates the point that it is difficult to assess reconstruction quality without considering how the data is to be used. In dealing with ordinary two dimensional images, it is often assumed implicitly that using the data means that a human being looks at it. With hyperspectral data, it is much more likely that human visual processing will be augmented or supplanted by automated processing. One might even go so far as to view hyperspectral data simply as an ensemble of one dimensional spectral signals, so that the concept of an "image" is irrelevant. In order to compare the different algorithms from this standpoint, we applied two spectrally based automatic processing algorithms to the reconstructed data. Although these algorithms may have limited practical utility by themselves, they are potential elements of more practical processing systems, and serve as useful illustrations.

The first algorithm classifies spatial resolution cells either as "object" (i.e., tent or house) or background cells based on the shape of their spectral profiles through the use of a simple Bayesian classifier as described in [10]. This approach was chosen for its simplicity and ease of interpretation. Although other, more powerful classifiers exist, we wanted to avoid clouding the compression evaluation with questions about the classifier. Also, this classifier is well known and was easily implemented through the use of the Khoros Image Processing system [11].

The classifier was designed in several steps. First, the image was preprocessed so that each spectra had unit energy. This was done so that the classifier made its decisions based on the shape of the spectra rather than on the overall intensity and would work equally well under different scene illuminations. Second, the image was clustered by the k-means clustering algorithm which is essentially the Lloyd-Max vector quantizer. Clustering is performed by first starting with an initial set of cluster centers and, at each iteration, assigning each data point to the nearest cluster center and then recomputing the cluster centers. Both the number of clusters and the initial set were chosen by hand so that representative samples from each class were included. Third, the clusters were assigned to classes by visually inspecting the image. The result of the classifier design was, for each data set, a set of clusters for each class and statistics (mean and covariance) for each cluster. Pixel by pixel classification is performed by finding the Mahalanobis distance to each cluster center (using the cluster mean and covariance) and finding the minimum. The class containing this cluster as a member is the class assignment for the data point.

We applied the classifier to the reconstructed data sets, and collected statistics on spatial cells that were classified differently in the original and reconstructed data sets. Figure 5 shows the percentages of "object" pixels in the original data misclassified in the reconstructed data as a function of compression ratio for each compression algorithm (the lines marked with o's). The same trends seen in the PSNR measurements are evident in this table: spectral decorrelation classified the most accurately, followed by 3-D wavelets, then the band-independent algorithm. The differences between algorithms are dramatic. The 3-D wavelets algorithm missclassifies about half as frequently as the band independent algorithm at similar compression ratios, and the worst

case spectral decorrelation algorithm performance is better than the best case for the other algorithms. Figure 6 shows maps of cell classifications for the original data sets and reconstructed data sets at the highest compression rate for each algorithm. Figure 7 shows corresponding maps of misclassified cells. All of these maps are at the lowest compression ratio tested for each algorithm. These results show that the classification algorithm is more sensitive to distortion than visual comparisons. At these relatively low compression rates, spectral band image distortion is not readily visible. Nonetheless, it induces significant classification errors.

The second example algorithm segments a complete hyperspectral data set into spatial regions such that cells within a region have similar spectral profiles. The segmentation process compares the spectral profile of the data at each spatial location to its neighbors; thus both the spatial and spectral properties of the data are important. Segmentations of the original cube and the compressed and uncompressed version are compared, both by visual inspection, and through a measure of differences between the edge maps. This measure combines discrepancies of two types: those where a pixel was marked as an edge in the original and not in the compressed/uncompressed data, and those where a pixel was marked as an edge in the compressed/uncompressed and not in the original. The two types of "errors" were combined to give a final measure of edge detection errors, expressed as a percentage of pixels across the entire image. While this error measure is simple, it is sufficient to provide a measure of the amount of distortion in the spatial and spectral properties of the cube.

The result of applying the spectral segmented to the "tents" data cube is shown in Figure 8. The boundaries of each region of the image are marked in dark. The resulting segmentations of applying the same algorithm to the compressed/uncompressed data sets using the three compression algorithms with three different compression ratios each are also shown in Figure 9. Quantitative measures of the edge errors for each of the three approaches (at various compression ratios) are shown for three different data sets in Figure 5 (the line labeled with x's). For all three cases, the spectral decorrelation algorithm produced segmentations closest to the original data, followed by the three dimensional transform approach.

References

- [1] T. Markos and J. Reif, Multispectral Image Compression Algorithms, Proceedings of 1993 Data Compression Conference, Mar. 1993, pp. 391-400.
- [2] J.N. Bradley and C.M. Brislawn, Applications of Wavelet-Based Compression to Multidimensional Earth Science Data, Proceedings of 1993 Data Compression Conference, Mar. 1993, pp. 496.
- [3] B.R. Epstein, R. Hingorani, J.M. Shapiro and M. Czigler, Multispectral KLT-Wavelet Data Compression for Landsat Thematic Mapper Images, Proceedings of 1992 Data Compression Conference, Mar. 1992, pp. 200-208.
- [4] R.L. Baker and Y.T. Tse, Compression of high spectral resolution imagery, Proceedings SPIE: Applications of Digital Image Processing, Aug. 1988, pp. 255-264.
- [5] B. W. Evans, Coding Quantizer Parameters for Wavelet Image Compression, TRW IOC G685.6-0301, Jun. 1992.
- [6] B. W. Evans and R. L. Renner, Optimal Bit Rate Allocation for Wavelet Transform Coding, Proceedings of 1993 Data Compression Conference, Mar. 1993, pp. 473.
- [7] B.W. Evans, Optimal Quantizer Bit Allocation for Wavelet Image Compression, TRW IOC G685.6.0294, Feb. 1992.
- [8] I. Daubechies, The wavelet transform, time-frequency localization and signal analysis, IEEE Trans. Information Theory, vol. 36, Sept. 1990, pp. 961-1005.
- [9] S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Acoustics, Speech and Signal Processing, Dec. 1990, pp. 2091-2110.
- [10] R. Duda and P. Hart, Pattern Classification and Scene Analysis, Wiley and Sons, N.Y., 1973.
- [11] Rasure, Williams, Agiro and Sauer, "A Visual Language and Software Development Environment for Image Processing", International Journal of Imaging Systems and Technology, Vol. 2, pp 183-199, 1990.



Figure 1. Single spectral band image from "houses" data set.



Figure 2. Single spectral band image from "tents" data set.

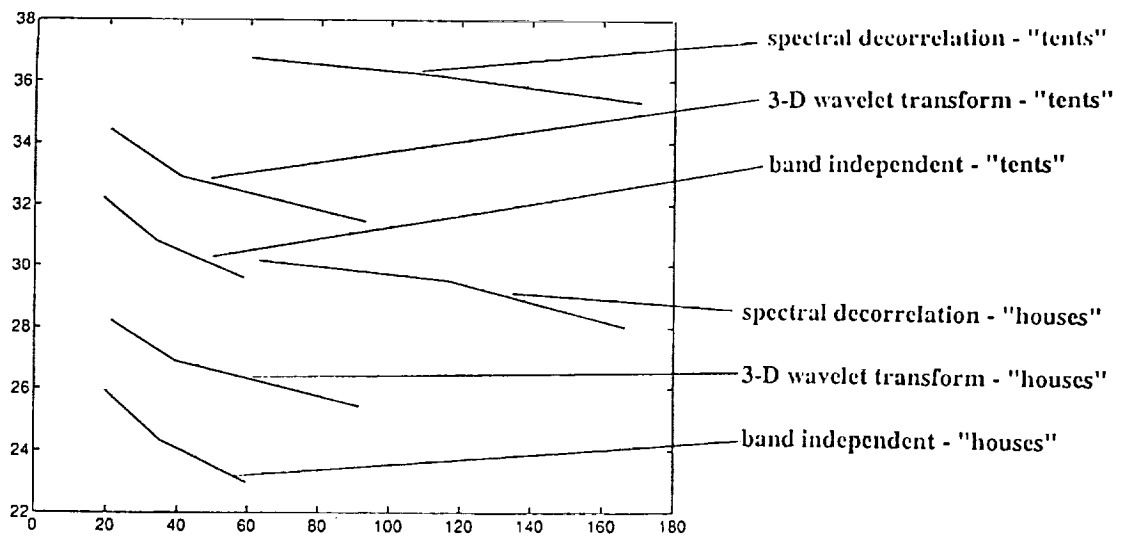


Figure 3. Peak signal to noise ratio vs. compression ratio.

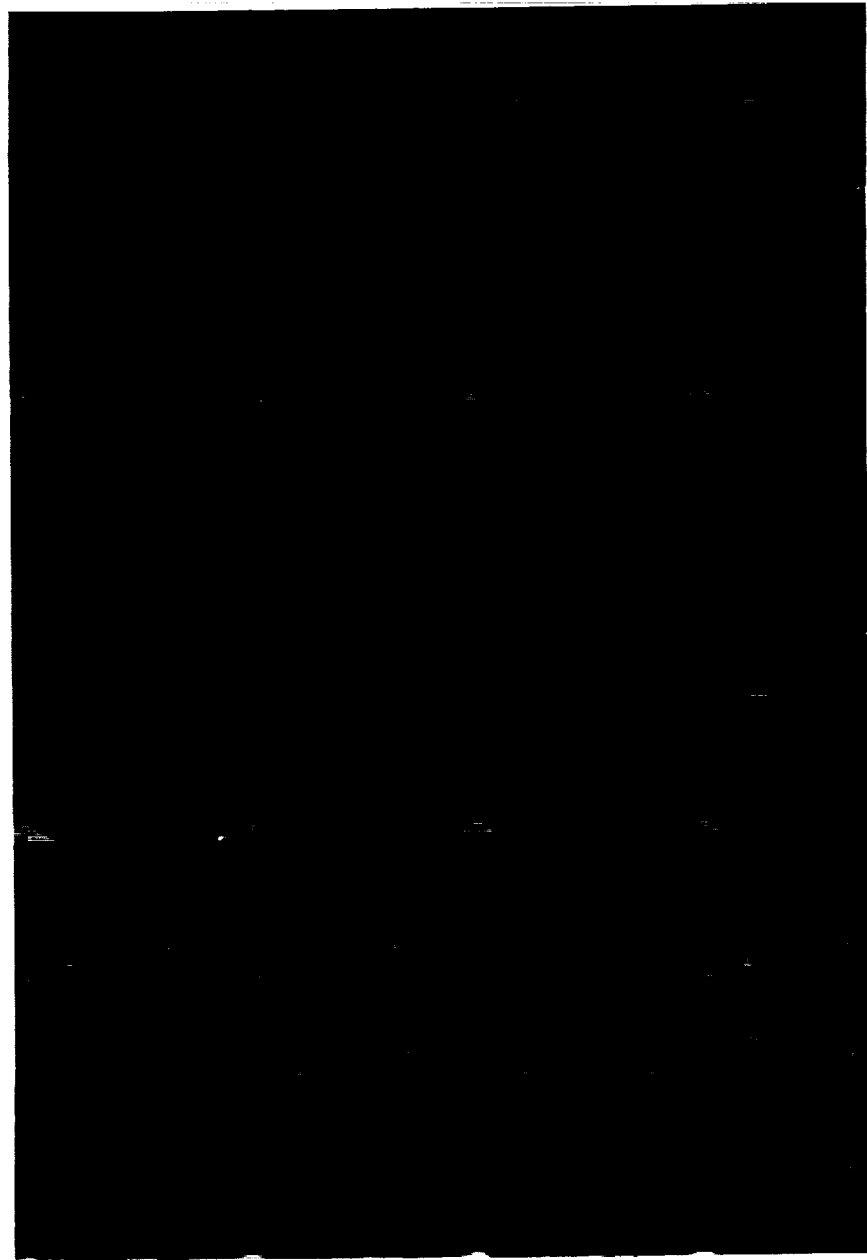


Figure 4. Examples of reconstructed images. Top row: band independent compression, from left to right: original image, 19:1, 34:1 and 59:1 compression. Middle row: 3-D wavelet compression, from left to right: original, 21:1, 41:1 and 92:1 compression. Bottom row: spectral decorrelation compression, left to right: original, 60:1, 112:1, 171:1 compression.

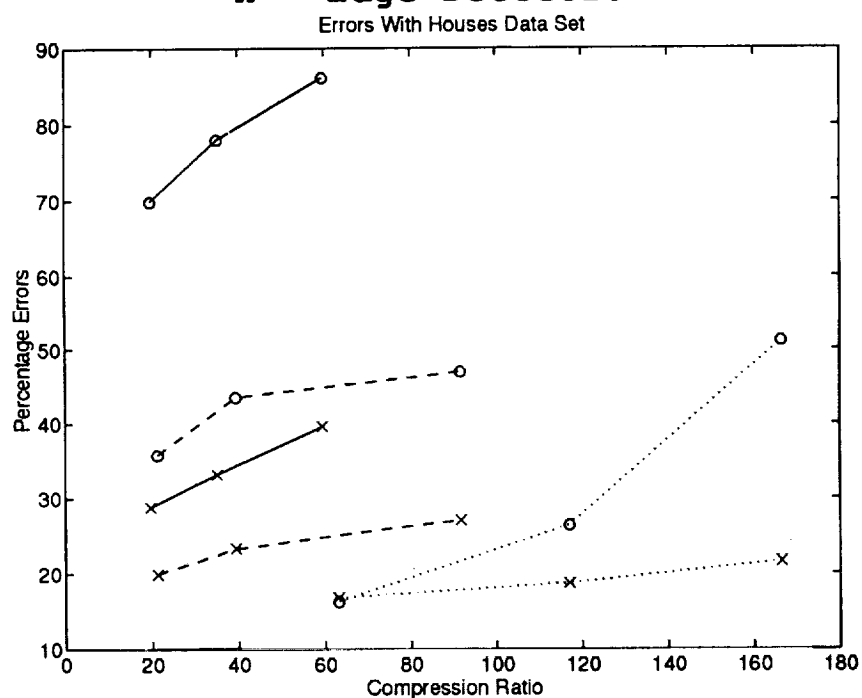
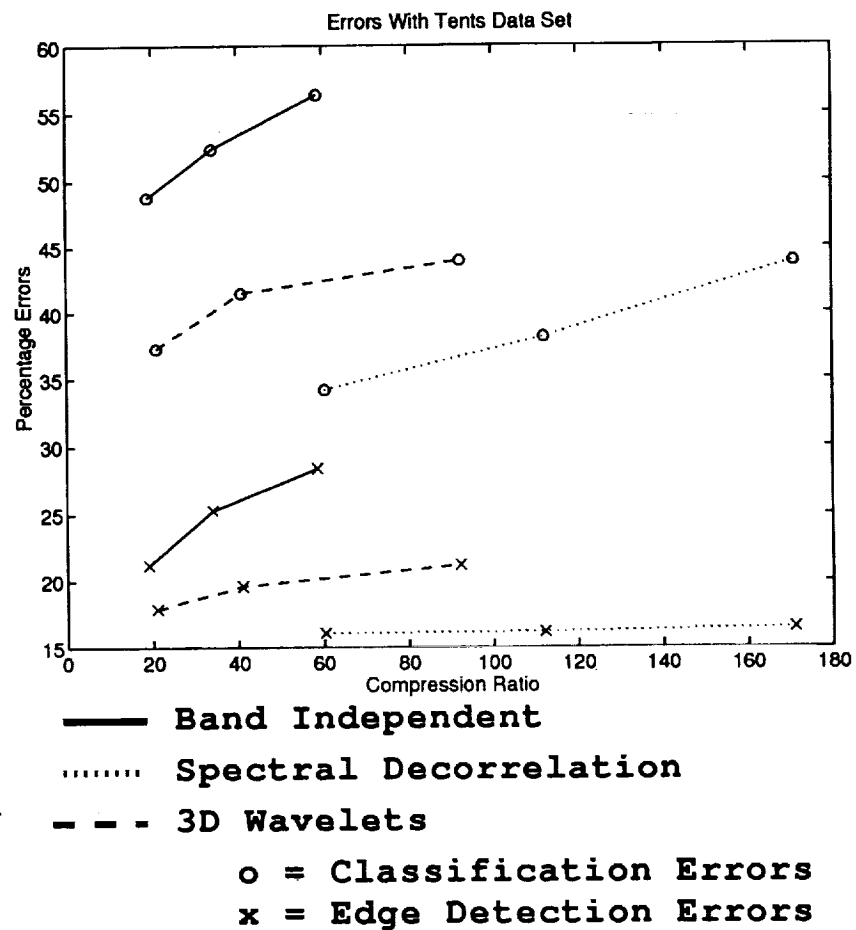


Figure 5. Percent of cells misclassified vs. compression ratio. Top graph: "tents" data set; bottom graph: "houses" data set.

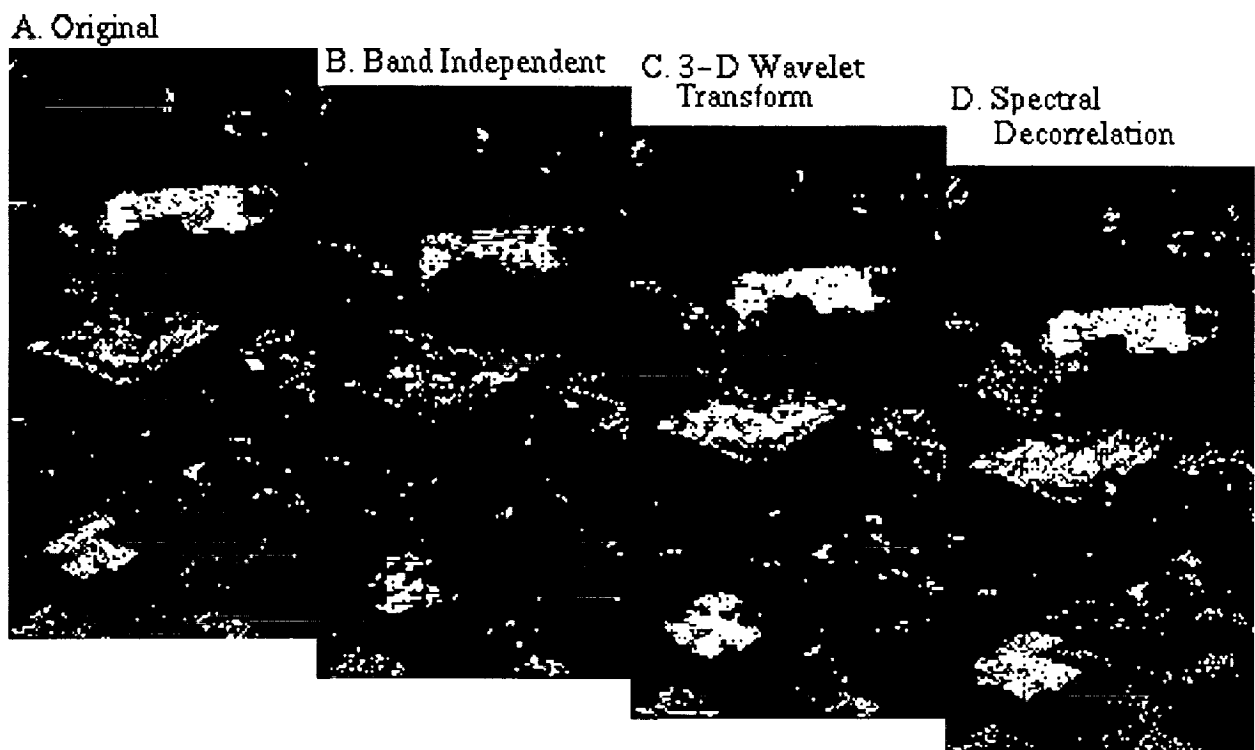


Figure 6. Cell classification maps. "Object" cells shown in white.

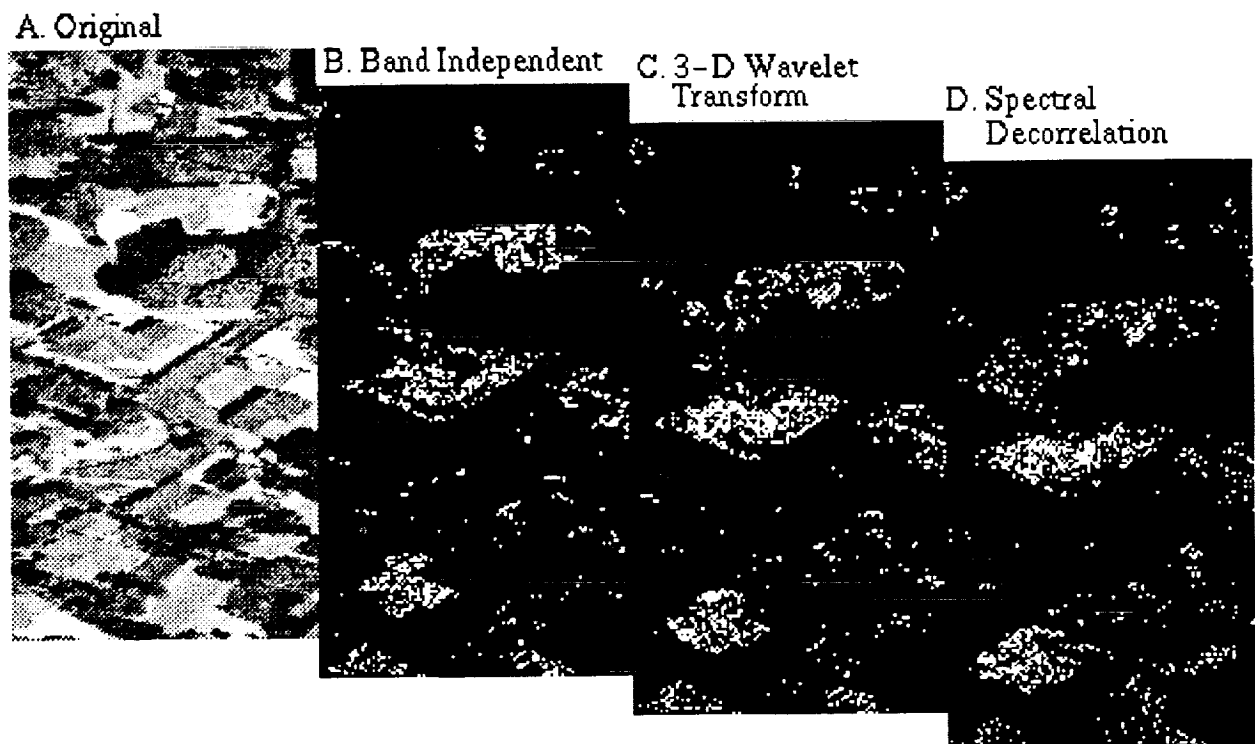


Figure 7. Misclassified cell maps. Incorrectly classified cells shown in white.



Figure 8. Region boundaries for original "tents" data set.

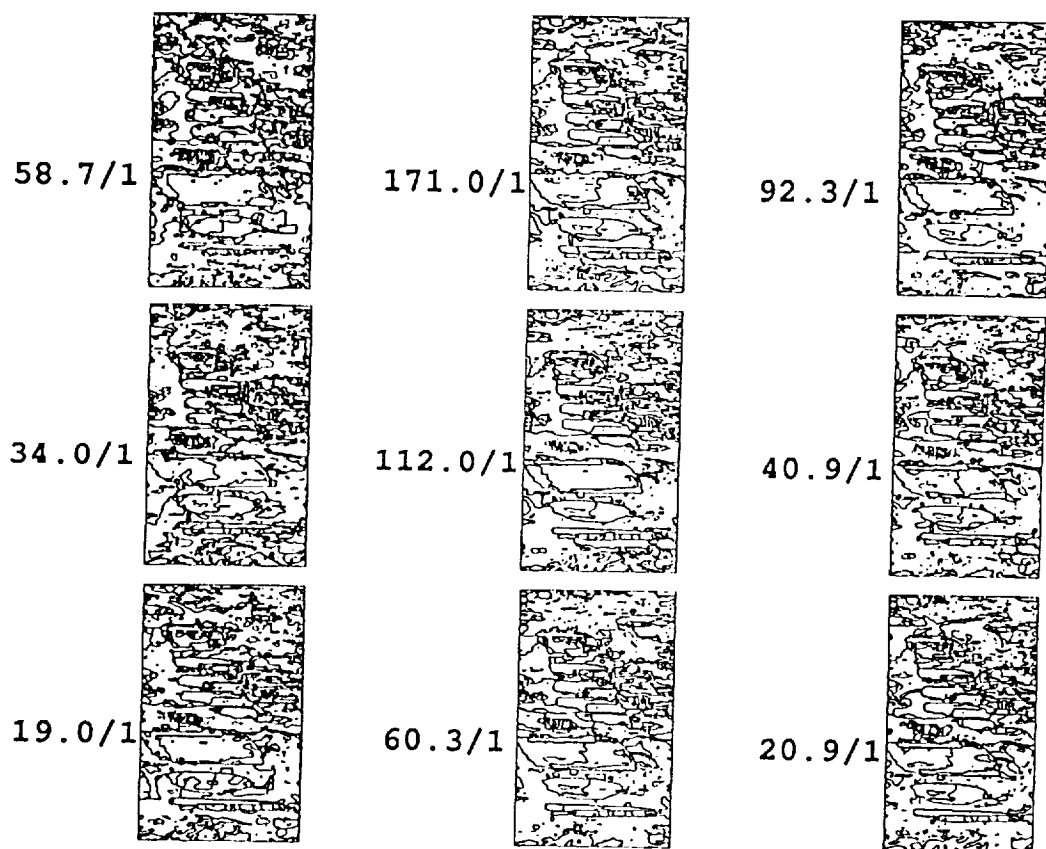


Figure 9. Region boundaries for reconstructed data sets. Left column: band independent algorithm. Middle column: spectral decorrelation algorithm. Right column: 3-D wavelet transform algorithms. Compression ratios shown to left of each image.

Some Practical Aspects of Lossless and Nearly-Lossless Compression of AVHRR Imagery

David B. Hogan¹, Chris X. Miller², Than Lee Christensen, Raj Moorti

Martin Marietta Astro-Space Division
Princeton, NJ 08543

Abstract

This paper evaluates compression of AVHRR imagery operating in a lossless or nearly-lossless mode. Several practical issues are analyzed including: variability of compression over time and among channels, rate-smoothing buffer size, multi-spectral preprocessing of data, day/night handling, and impact on key operational data applications. This analysis is based on a DPCM algorithm employing the Universal Noiseless Coder, which is a candidate for inclusion in many future remote sensing systems. It is shown that compression rates of about 2:1 (daytime) can be achieved with modest buffer sizes (≤ 2.5 Mbytes) and a relatively simple multi-spectral preprocessing step.

Introduction

Incorporation of compression into a real-time remote sensing system adds a number of complications. Lossless compression, desired by many users, necessarily results in a variable rate output. A rate smoothing buffer is thus required to interface to systems which require a fixed rate input such as real-time downlinks and magnetic tape mass storage. Also, since the possibility of buffer overflow cannot usually be eliminated, some means must be incorporated to reduce the rate below that achieved by lossless compression in such situations. Coding delay may also be an issue for real-time downlinks depending on the size of the buffer.

Martin Marietta Astro-Space Division has developed a test-bed consisting of both hardware and software to investigate such issues. The test-bed consists of: (1) a wide variety of compression algorithms (including both industry standard algorithms such as the Universal Noiseless Coder, the Joint Photographic Experts' Group discrete cosine transform algorithm and internally developed algorithms); (2) system modeling software such as rate smoothing buffers; and (3) diagnostic software to characterize compression algorithm performance and develop appropriate metrics. Most of the compression algorithms are implemented in a workstation environment. A number of algorithms are implemented on a real-time programmable signal processor. In this study, the test-bed was applied to investigate lossless compression of the Advanced Very High Resolution Radiometer (AVHRR) which flies on the TIROS series of low-altitude weather satellites.

AVHRR Data Set

A data set consisting of real-time AVHRR data acquired from the NOAA 11 and 12 satellites was assembled. The data were received at a High Resolution Picture Transmission (HRPT) Receiving Station which is part of the Advanced Remote Sensing Laboratory at Martin Marietta Astro-Space Division in Princeton, New Jersey. Both day and night passes were

¹ Current affiliation: Atmospheric and Environmental Research, Inc., 840 Memorial Drive, Cambridge, MA 02139-3794, Phone: 617-547-6207, Fax: 617-661-6479, E-Mail: dhogan@aer.com

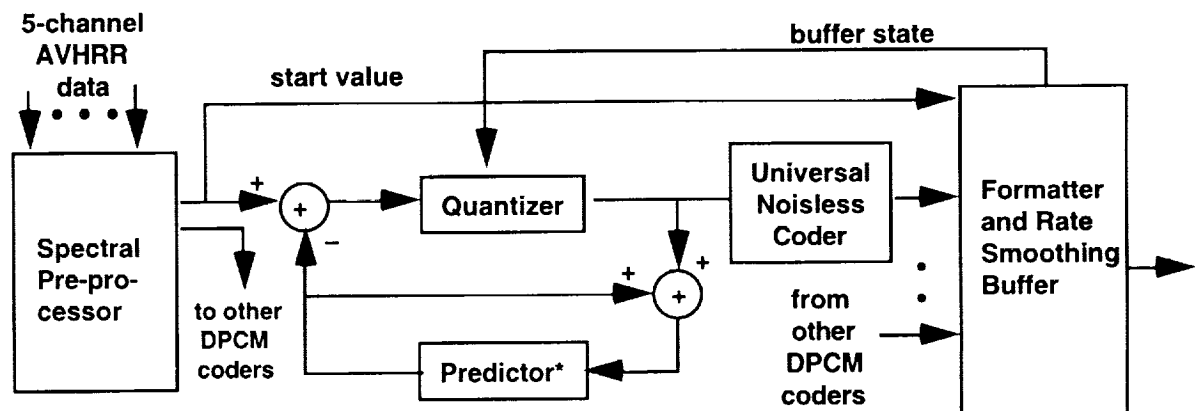
² Address: Martin Marietta Astro-Space Division, MS-410-1B, PO Box 800, Princeton, NJ 08543, Phone: 609-490-4510, Fax: 609-490-3962, E-Mail: cxm@saturn.astro.ge.com

assembled consisting of ~60 minutes of daytime data and ~57 minutes of nighttime data acquired from the ground station located in Princeton. A typical pass duration was 8-10 minutes. The data set covers a variety of regions and scene complexities, including ocean and land over latitudes ranging from 25°N to 55°N. Total data set size was about 860 Mbytes — somewhat greater than the data from one complete orbit.

The uncalibrated HRPT data were used in the analyses that follow. These data are for five bands (two in the visible/near infrared, one mid-wave infrared and two long-wave infrared) and have a spatial resolution of about 1.1 km at nadir for ~2,048 samples per scan line. Each sample is quantized to 10-bits. The HRPT data stream also contains sensor calibration samples, spacecraft telemetry data, frame synchronization and other miscellaneous headers, and data from lower rate sensors. Compression of these other data was not investigated. Total data rate of the HRPT stream is 666 kbits/s.

Compression Approach

A Differential Pulse Code Modulation (DPCM) coder followed by an entropy coder was used as illustrated in Figure 1. Both one- and two-dimensional (1-D and 2-D) predictors were tested. A simple three-point, 1-D predictor was used for most of the results reported here. 1-D predictors minimize front-end buffering and simplify error propagation control. Entropy coding was based on the Universal Noiseless Code (UNC) described by Rice (1991). The UNC was selected for several reasons: competitive performance when compared to other entropy coders for the type of data used in this study; anticipated availability in high-speed, rad-hard chips; and its inclusion in the Consultative Committee for Space Data Systems (CCSDS) standard for Advanced Orbiting Systems, Networks and Data Links (Yeh, et al, 1992).



Note: * DPCM predictor may alternately feedback to spectral preprocessor as described in text which would modify diagram (not shown)

Figure 1 DPCM Model Block Diagram

The UNC implementation employed eight of the alternative Rice coders $\Psi_{1,0}$ through $\Psi_{1,6}$ plus the default coder Ψ_3 . In Rice's nomenclature this translates to a coder with values $\lambda = 1$ and $N = 8$. No additional coding of the coder identifier was performed. We experimented with a variety of block sizes (J in Rice's nomenclature) and determined that $J = 16$ or $J = 32$ were near optimum for most cases. The starting values for the DPCM predictor were provided only once per scan line.

When operated in a lossless mode the quantizer of Figure 1 is the identity function. A uniform quantizer was used for lossy operation, as described later.

Although most of the experiments described here were performed on Sun SPARC-2 and SPARC-10 workstations, these algorithms have also been implemented on a real-time programmable signal processor developed by Martin Marietta built around the Texas Instrument TMS-320C30 chips. Rates in excess 1.5 Mpixels/s have been demonstrated on a four-processor version. Such a system may be an alternative to firmware solutions for moderate rate applications desiring flexibility and reprogrammability.

Some special procedures were added for nighttime data. While there is essentially no information in channels 1 and 2 at night for normal conditions (they measure reflected solar radiation), it is possible that such data might be of use for unusual circumstances. For this reason the channels were not completely eliminated in the final formatted product. Rather, at night the signal level which consists of the zero level plus random noise was replaced by a fixed value (in this case zero) when the signal is within some range determined by the expected noise level. This function could be implemented outside the UNC chip. This method provides a very high compression ($>>20:1$) for these channels but would still acquire rare special events at night with negligible impact to the overall performance.

Multi-spectral Preprocessing

It has long been recognized that spectral correlations among sensor bands can be used to further improve compression of multispectral data. However, since this decorrelation adds to the complexity of the system, its marginal benefit must be carefully weighed. In the case of the AVHRR, this improvement has been found to be small, but perhaps significant in some applications. Miettinen (1992) using a Discrete Cosine Transform (DCT) spatial compressor preceded by a Karhunen-Loueve spectral transform (KLT) found an 18% reduction in rate compared to spatial compression only for a fixed mean squared error (mse) at moderate compression ratios (8:1 to 15:1) but at low compression ratios and low mse ($\text{mse} < 1$ digital numbers per band and compression ratio $\leq 6:1$), the incremental benefit was less than 8%. As lossless performance is approached, this benefit is further reduced.

Among AVHRR bands, numbers 4 and 5 have the highest correlation (in excess of 95%). Both measure thermally emitted radiation in the 10-12 μm window region with most of the brightness temperature differences (ΔT_B almost always less than 2 K) arising from small differences in water vapor absorption (for scenes viewing the surface). Thin cirrus (ice) clouds have been shown to likewise result in a small but significant signature in ΔT_B . The compression of each channel individually was compared to sending bands 1 through 4 plus the difference of bands 4-5. For lossless compression, a reduction in data rate of 5.5% was achieved when averaged over all bands (reduction from 5.25 to 4.96 bits per pixel per band, bpppb).

The final algorithm also employed the differences of bands 1 and 2 which reduced the rate another few percent. No spectral preprocessing was applied to band 3 ($\sim 3.7 \mu\text{m}$) which responds to both thermally emitted radiation and reflected solar radiation during the day and shows only modest correlation with the other bands. This is probably due to a combination of the more complex phenomenology and the excess sensor noise often experienced by this channel.

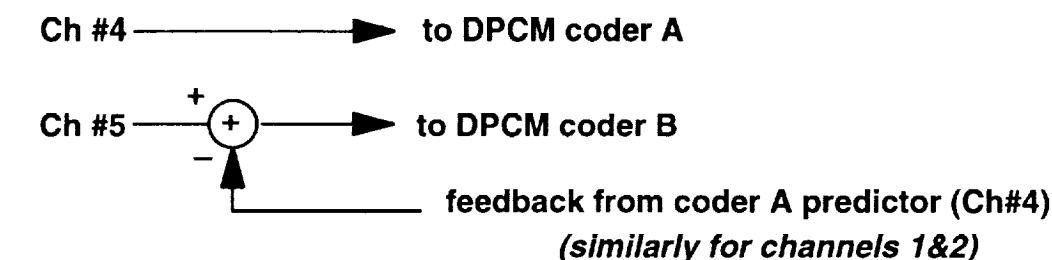
An additional modification must be made to allow a lossy mode. One possibility is to include the spectral preprocessor in the DPCM feedback loop (see Figure 2a). While not inherently difficult to implement it does add to the complexity of the spectral and spatial compressor interface. An alternative is to send both the difference and sum of the bands (Figure 2b). As any errors introduced by the quantization step are now orthogonal, no feedback is necessary. The reader will undoubtedly recognize this as the degenerate case of the KLT for two bands (without the scaling) — the only KLT which is data independent. Figure 2c illustrates a five-band orthogonal spectral preprocessor. As long as the KLT

transform vectors are prestored (i.e., calculated on the ground) and not calculated in real-time, this presents only a modest computational burden.

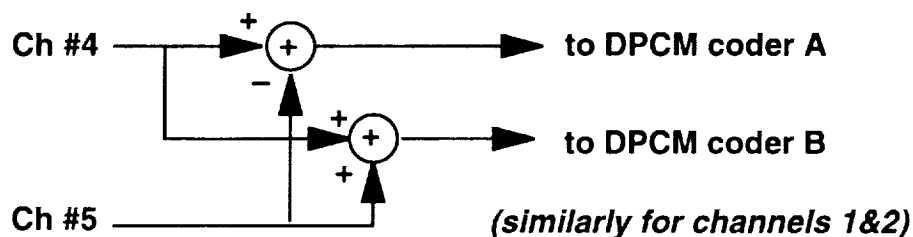
Thus, it has been found that an optimal five-band spectral transform (i.e., KLT) is not necessary to secure most of the advantage from spectral correlations for a multispectral compressor. Operating on differences between bands 1 & 2 and 4 & 5 has the added benefit that several of the key applications of AVHRR data employ these channel differences in a rather direct way (e.g., sea surface temperature and normalized difference vegetation index). This naturally leads to methods for optimizing the compression algorithm for user processing.

Rate-Smoothing Buffer Sizing

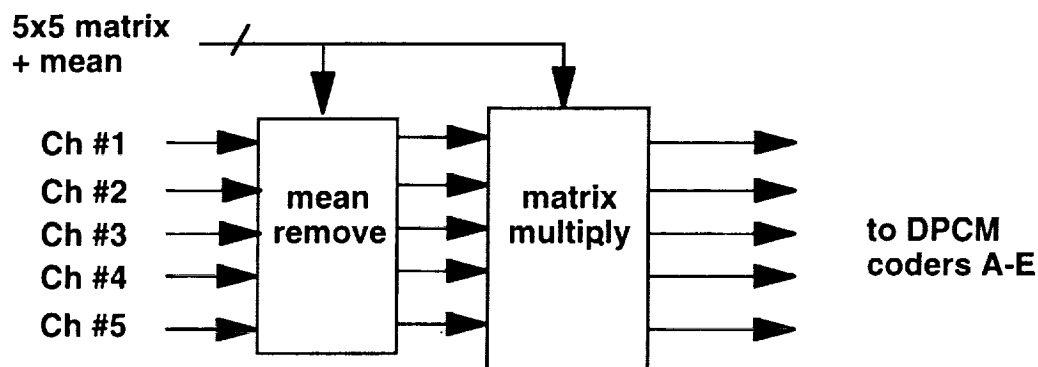
A model was developed which emulated the system of Figure 1. The following parameters are specified for the rate smoothing buffer: buffer size (in bytes), initial buffer state (percent full), and fixed output rate. The day and night AVHRR data were then separately processed by the model. Statistics were kept for the fraction of time the rate smoothing buffer was maintained in various states of fullness.



(a) Spectral DPCM Preprocessor



(b) Orthogonal Spectral Preprocessor



(c) Five-band Orthogonal Spectral Preprocessor

Figure 2 Alternative Spectral Preprocessors

Figure 3 gives an example of the variable compression ratio (averaged over single AVHRR scan lines) versus scan line number for a typical pass. The compression variability, ranges from 6.6 bpp to 4.8 bpp. The very low rates are communication drop-outs experienced by the receiving station.

Figure 4a shows a histogram of buffer state for the daytime data set with a 2.5 Mbyte buffer and a 4.9 bpp fixed rate output. The buffer was in an overflow state approximately 1% of the time. These conditions can be handled by one of several approaches discussed in the next section. Figure 4b shows a similar histogram for the nighttime data set.

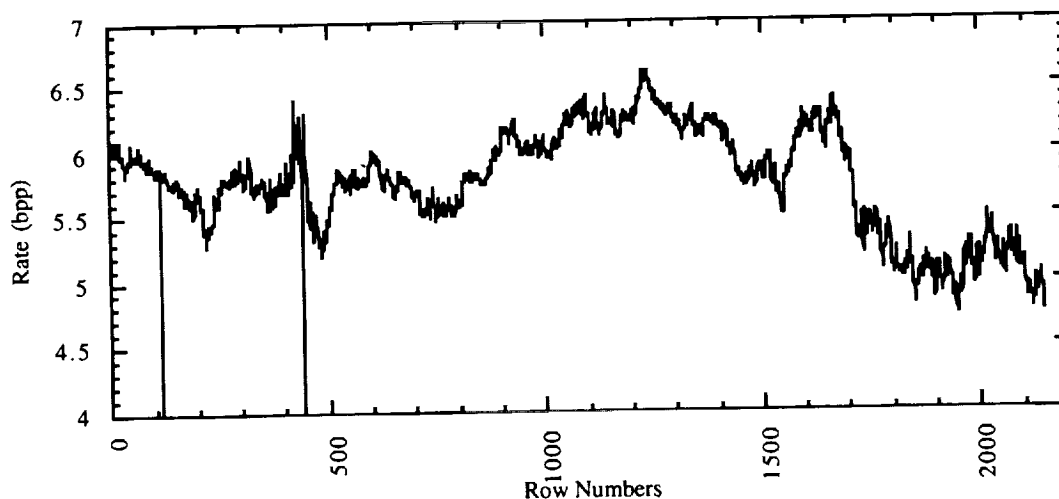
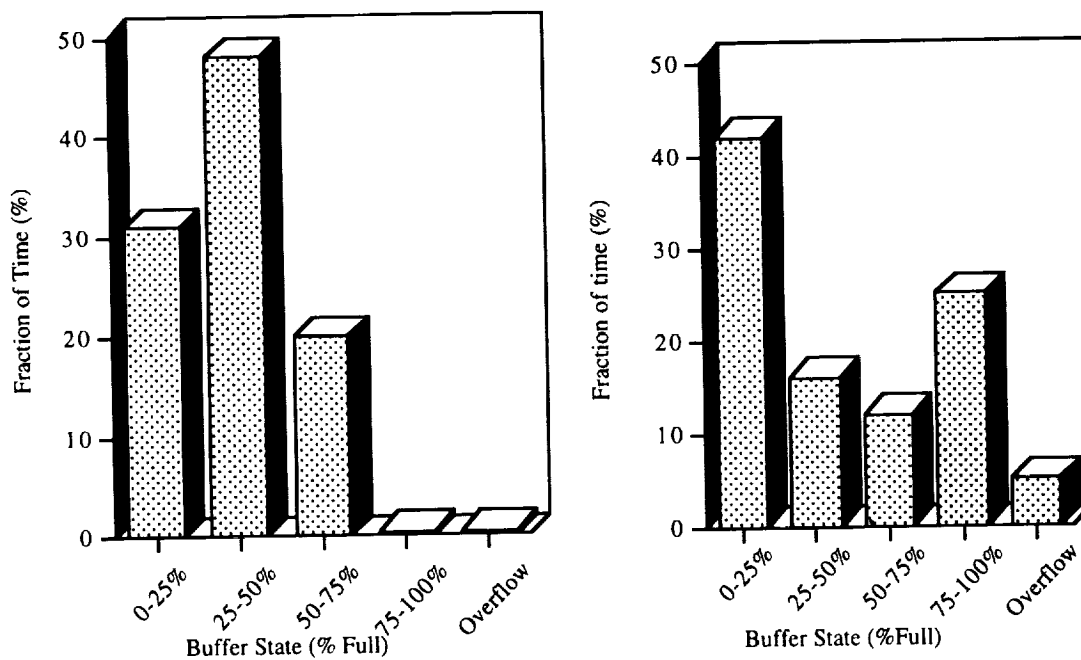


Figure 3 Line-averaged Lossless Compression Rate for a Typical AVHRR Pass



(a) Daytime Data Set

(b) Nighttime Data Set

Figure 4 Buffer State Histograms

Next the fixed out rate was varied with a fixed buffer size. Figure 5a plots the fraction of data overflowed in the buffer versus the fixed output rate for buffer sizes of 1 and 2 Mbytes. These results suggest that a buffer size of 2 Mbytes corresponding to about 3.5 minutes of data is sufficient to operate losslessly all the time at a fixed output rate of 5.1 bpppb. Similar calculations with nighttime data (Figure 5b) indicate that a rate of 3 bpppb (averaged over 5 bands) can be achieved with a 5 Mbyte buffer. A smaller buffer only increases the amount of buffer overflow by a small amount ($\ll 1\%$ for 1.0 Mbyte buffer).

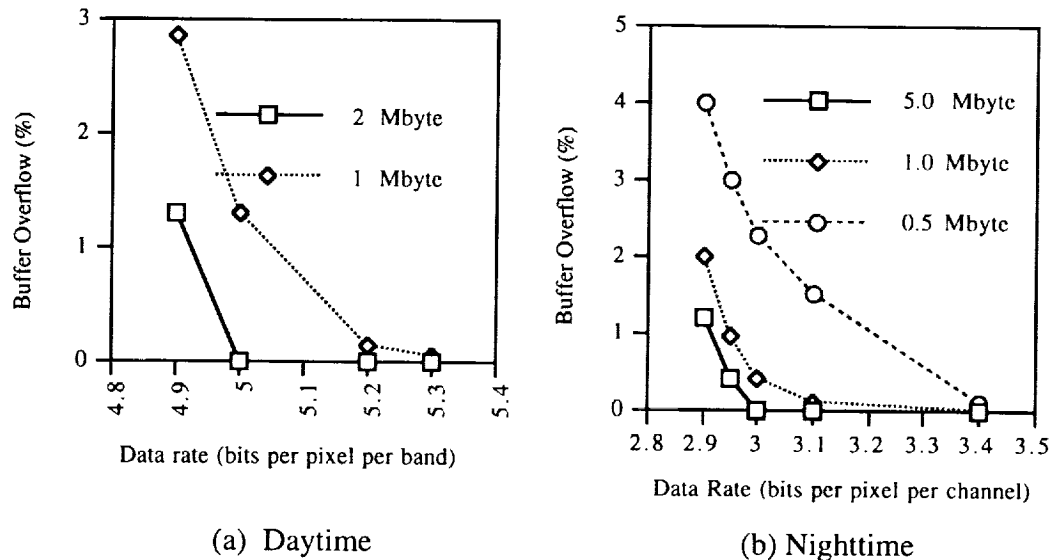


Figure 5 Buffer Overflow versus Output Rate for Various Buffer Sizes

Graceful Degradation Mode

Rice (1991) describes several methods for adapting the UNC to a lossless mode. The leading candidates described are truncation at the edge of the scan and progressive elimination of low order bits. The former method is reasonable for planetary missions where a camera is centered on a target of interest (typical of planetary missions for example). It is less reasonable for a system such as the AVHRR where global coverage and continual monitoring are desired. In the second method, the elimination of high order bits can be facilitated by an appropriate ordering of the UNC output stream. This method provides all the data and the loss can be selectively applied (for example to lower priority regions). A number of implementation variants are also described such as a zig-zag ordering method which may offer an advantage for some applications.

For this paper, a third approach is used which provides the rate control feedback through the quantizer. A uniform quantizer is used which has been shown to provide nearly optimum performance — in terms of its rate distortion function — for a scalar quantizing system using entropy coding of a memoryless source (Farvardin and Modestino, 1984).

Some trades of rate versus distortion for the uniform quantizer are shown in Figure 6. The rate is reduced from 5.5 bpppb lossless to 3.8 bpppb with an mse of $\sim 0.6 \text{ DN}^2$ (digital numbers). Thus, significant control of output rate can be achieved with very modest errors introduced to the data. This distortion plateaus near an mse of 0.5 DN^2 due to the 10-bit quantization of the data input to the quantizer. By maintaining more bits precision in the multi-pixel predictor (or preferably in the original sensor data), rounding problems with the uniform quantizer can be minimized. The rate-distortion curve would then exhibit a more gradual degradation.

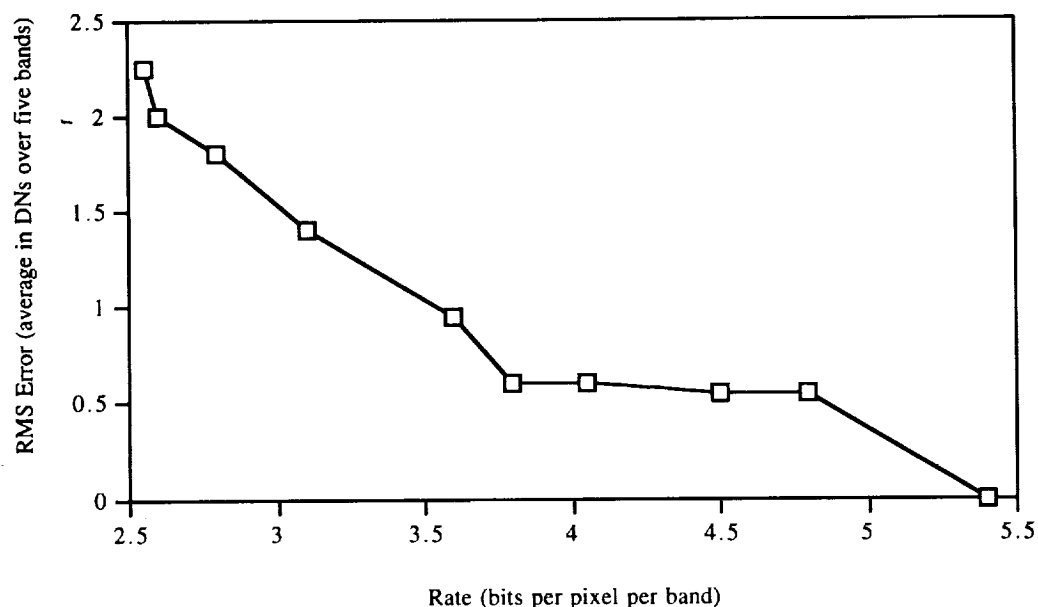


Figure 6 Rate Distortion Function for Uniform Scalar Quantizer

The proposed rate control method consists of determining a parameter T , the lossless/lossy buffer fullness threshold, and a function $r(S)$, the quantizer feedback. As long as the buffer state of fullness $S \leq T$, the system operates in a lossless mode. When $S > T$, the uniform quantizer is supplied with a divisor determined by $r(S)$. Further experiments are required to determine the optimal T and $r(S)$. It appears that a linear function will be adequate.

Discussion

A model has been developed for evaluating lossless compression performance using the Universal Noiseless Coder and applied to the AVHRR. A variety of system parameters can be traded using this model such as buffer size, fixed output rate, etc. It has been determined that a strictly one-dimensional compressor using a 3 point predictor can achieve compression from the original 10-bit AVHRR data to ~5 bits per pixel per band for daytime and ~3 bits per pixel per band for nighttime with buffer sizes less than 2 Mbyte. The results summarized in Table 1 indicate that even for the nearly-lossless mode, that maximum errors of < 1 DN. The corresponding mean square errors would be $\ll 1$.

Table 1 Lossless and Nearly-Lossless Compression Summary

Mode	Rate (bpp)	Buffer (MB)	Lossy Fraction*	Max. error (DN)**
Day:				
Lossless	5.1	2.0	$< 1\%$	1
Nearly-lossless	4.9	1.0	$\sim 4\%$	1
Night:				
Lossless	3.0	2.0	$< 1\%$	1
Nearly-lossless	2.8	1.0	$\sim 4\%$	1
Notes: * Fraction of time spent is lossy mode				
** Estimated maximum error during lossy mode, mse < 0.5				

The 1-D compressor described has the advantage that any bit stream errors cannot propagate beyond the line in which they occur. The maximum coding delay of 3.5 minutes is not expected to be significant for most situations. A simple sum/difference spectral preprocessor

applied to channels 1&2 and 4&5 respectively was shown to provide a small but potentially useful reduction in rate (6-8%) when compared with compressing each channel independently.

While the system parameters above can provide lossless performance the vast majority of the time, buffer overflow might still occur. A feedback system to a uniform quantizer was recommended and examples of the rate distortion function were given. Should a fixed output rate near the average lossless rate be selected, this could provide a fallback mode for rare circumstances when the buffer overflows. Since errors are small — less than the inherent noise level of the sensor — the impact on data quality would be very small. Should a rate below the lossless average rate be desired, the rate distortion function suggests that $mse < 1 DN^2$ can be achieved with rates up to 2 bpp below the lossless rate.

While silicon implementations of the UNC are available (Yeh, et al, 1992), additional support circuitry would be required in any event to perform the spectral and spatial predictions and to implement the quantizer. An alternative is to employ programmable signal processors. This adds considerably to the flexibility of the compressor. Minor and possibly major modifications to the algorithm could be made even during a mission. The UNC has been tested in such a system at Martin Marietta. The programmable signal processor uses four Texas Instrument TMS320C30 processor supplemented by custom interface chips to enhance interprocessor communications. The UNC algorithm, using a somewhat simpler predictor than described here, has been benchmarked at rates in excess of 1.5 Mpixels/s on this system. This is much greater than the ~60 kpixels/s rate at which the AVHRR operates.

Future work will expand the model in a number of ways. The graceful degradation mode will be integrated with the overall model. Ability to analyze the impact of bit stream errors will also be incorporated. Furthermore, radiometrically critical AVHRR applications such as Sea Surface Temperature (SST) and Normalized Difference Vegetation Index (NDVI) will be investigated. Additionally, greater quantities of data will be tested and other multispectral sensors will be considered.

Acknowledgment

This work was performed entirely at Martin Martin Astro-Space Division as part of their Internal Research and Development Program.

References

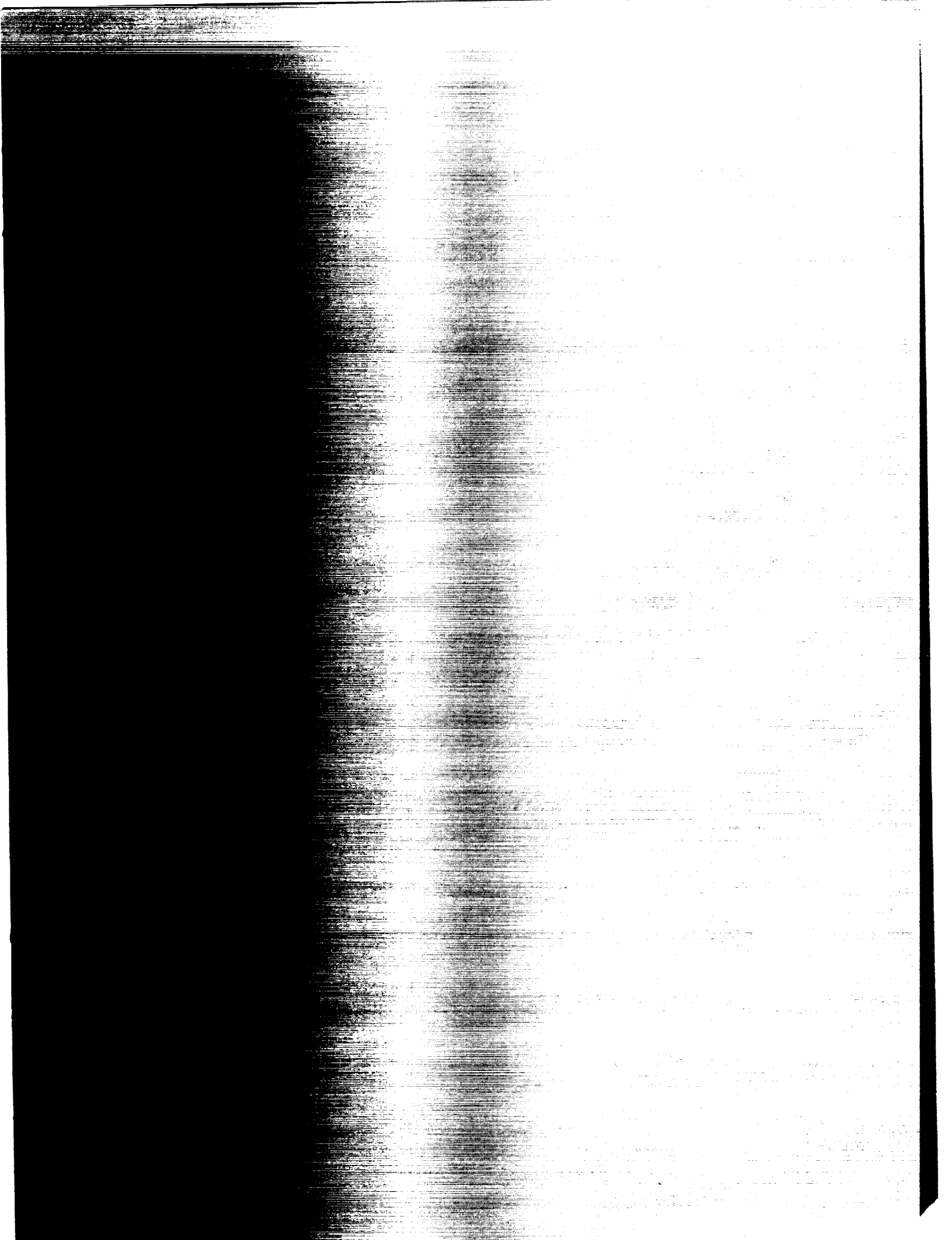
- Farvardin, Nariman and James Modestino, "Optimum Quantizer Performance for a Class of Non-Gaussian Memoryless Sources," IEEE Transactions on Information Theory, IT-30(3), May 1984, pp. 485-497.
- Miettinen, Kristo, "Compression of Spectral Meteorological Imagery," Space and Earth Science Data Compression Workshop, Snowbird Utah, March 1992.
- Rice, Robert, "Some Practical Universal Noiseless Coding Techniques Part III, Module PSI14,K+," JPL Publication 91-3, NASA Jet Propulsion Laboratory, Pasadena CA
- Yeh, Pen-Shu, Warner Miller, Jack Vanbrux, Norley Liu and Robert Rice, "The Implementation of a Lossless Data Compression System Module in an Advanced Orbiting System," Space and Earth Science Data Compression Workshop, Snowbird, Utah, March 1992.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

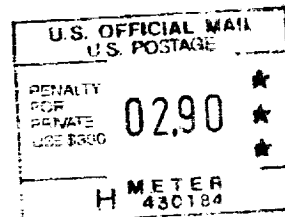
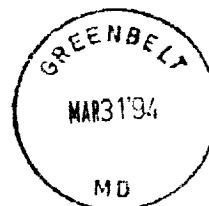
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 1994		3. REPORT TYPE AND DATES COVERED Conference Publication	
4. TITLE AND SUBTITLE Space and Earth Science Data Compression Workshop				5. FUNDING NUMBERS 930	
6. AUTHOR(S) James C. Tilton, Editor					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Goddard Space Flight Center Greenbelt, Maryland 20771				8. PERFORMING ORGANIZATION REPORT NUMBER 94B00043	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001				10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CP-3255	
11. SUPPLEMENTARY NOTES This workshop was organized and sponsored by the National Aeronautics and Space Administration (NASA) and the Center of Excellence in Space Data and Information Sciences, in cooperation with the 1994 Data Compression Conference (DCC '94).					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 59				12b. DISTRIBUTION CODE 1	
13. ABSTRACT (Maximum 200 words) This document is the proceedings from the fourth annual "Space and Earth Science Data Compression Workshop," which was held on April 2, 1994, at the University of Utah in Salt Lake City, Utah. This workshop was held in cooperation with the 1994 Data Compression Conference (DCC '94), which was held at Snowbird, Utah, March 29-31, 1994. The Workshop explored opportunities for data compression to enhance the collection and analysis of space and Earth science data. It consisted of thirteen papers presented in four sessions. The papers focus on data compression research that is integrated into, or has the potential to be integrated into, a particular space and/or Earth science data information system. Presenters were encouraged to take into account the scientist's data requirements, and the constraints imposed by the data collection, transmission, distribution and archival system. The workshop was organized by James C. Tilton of the NASA Goddard Space Flight Center, Sam Dolinar of the Jet Propulsion Laboratory, Sherry Chuang of the NASA Ames Research Center, and Dan Glover of the NASA Lewis Research Center.					
14. SUBJECT TERMS Data Compression, Image Compression, Signal Processing, Image Processing, Space Science, Earth Science				15. NUMBER OF PAGES 142	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	
20. LIMITATION OF ABSTRACT Unlimited					



National Aeronautics and
Space Administration
Code JTT
Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, \$300



S1 002 CP-3255 940131 S090569 A
NASA
CENTER FOR AEROSPACE INFORMATION
ACCESSIONING
800 ELKRIDGE LANDING ROAD
LINTHICUM HEIGHTS MD 210902934

